AD-A093 252 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/3

ADVANCED SIMULTION OF DIGITAL FILTERS.(U)

SEP 80 G S DOYLE

UNCLASSIFIED NL

1 OF 4

AD

A093252

LEVEL
2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD A093252

DTIC
ELECTE
DEC 3 0 1980
C

THESIS

ADVANCED SIMULATION OF DIGITAL FILTERS

by

Gerald S. Doyle

September 1980

Thesis Advisor: D. E. Kirk

Approved for public release; distribution unlimited

80 12 29 169

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	
2. GOVT ACCESSION NO.	AD-A093 252
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	Advanced Simulation of Digital Filters
5. TYPE OF REPORT & PERIOD COVERED	Master's Thesis; September 1980
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	Gerald S. Doyle
8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Naval Postgraduate School, Monterey, California 93940
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	Naval Postgraduate School, Monterey, California 93940
12. REPORT DATE	September 1980
13. NUMBER OF PAGES	307
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	
15. SECURITY CLASS. (of this report)	UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Filter; Interactive Graphics; Z-Plane

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An Advanced Simulation of Digital Filters has been implemented on the IBM 360/67 computer utilizing Tektronix hardware and software. The program package is appropriate for use by persons beginning their study of digital signal processing or for filter analysis. The ASDF programs provide the user with an interactive method by which filter pole and zero locations can be manipulated. Graphical output on both the Tektronix graphics screen and the Versatec plotter are provided to observe the effects of pole-zero movement

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

Advanced Simulation of Digital Filters

by

Gerald S. Doyle
Captain, United States Army
B.S., United States Military Academy, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September, 1980

Accession For
NTIS
DTIC
Unannounced
Justification
By
Distribution/
Availability
Avail
Dist
Special
A

Author: Gerald S Doyle

Approved by: D E Kirk
Thesis Advisor

R D Strum
Second Reader

D E Kirk
Chairman, Department of Electrical Engineering

William M. Tolles
Dean of Science and Engineering

ABSTRACT

An Advanced Simulation of Digital Filters has been implemented on the IBM 360/67 computer utilizing Tektronix hardware and software. The program package is appropriate for use by persons beginning their study of digital signal processing or for filter analysis. The ASDF programs provide the user with an interactive method by which filter pole and zero locations can be manipulated. Grpahical output on both the Tektronix graphics screen and the Versatec plotter are provided to observe the effects of pole-zero movement.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author wishes to acknowledge the people who made this effort possible. Of greatest importance is the love and guidence provided by his parents, Margaret and C. E. Doyle, and sister, Evelyn. It is to them that this thesis is dedicated. Special mention goes to Professor Richard Hamming for his insight and explanations of finite precision effects in digital filters. Thanks are due to Professors D. E. Kirk and R. D. Strum for their suggestions and comments in structuring this work. For their assistance with system problems, thanks are due to J. Foust, F. Wheeler, R. Donat, and H. Doleman. Moreover, thanks are due to E. Christian, and M. F. Bradley for their assistance in preparing the manuscript. Of no less importance are the evening computer center operators J. Kallweit, M. Anderson, and E. Donnellan for their consideration and assistance in processing the figures. However, the author alone is responsible for any shortcomings in this thesis, for the ultimate choice of what was done was his.

I. INTRODUCTION

Digital methods have recently begun to dominate the selection of transmission and filtering schemes. When the student first confronts the discrete domain, he encounters a series of unfamiliar concepts. While most of the discrete concepts have parallels in the continuous or s-domain, this dualism is not apparent initially.

The programs comprising the Advanced Simulation of Digital Filters (ASDF) provide the user with a convenient method for developing intuition in the discrete domain. This is achieved by allowing the user to generate digital filters by directly entering the poles and zeros into a graphics terminal. The ASDF programs then perform the calculations required to display the characteristics of the user generated filter.

The overriding objective of the ASDF program package is to provide the user with an interactive method for modifying the filter pole/zero locations in the z-plane, as well as to compute the filter responses. Both time and frequency responses are computed and displayed graphically on the Tektronix 4012 graphics terminal. [Ref. 9]

II. SELECTION OF ASDF ALGORITHMS

There are several parameters which are particularly important in selecting the algorithms used in the ASDF programs. Since one of the primary objectives of these programs is to provide the user with clear explanations of methods for computing digital filter characteristics, direct or straightforward methods should be used. The importance of developing a user's intuition for the discrete domain and digital filters suggests that graphical rather than numerical outputs would be most appropriate. The use of graphical outputs to characterize a filter's responses implies minimum importance of precision in these results. Since the reason for computing the filter characteristics is to develop the user's intuition for the discrete domain, there are no pressing constraints on the selected algorithms with regard to computational efficiency, speed, or memory space utilization. In fact, the final ASDF package spends the greatest portion of its execution time making the interface with the z-plane, and the form of the output easy to understand and to use. The greatest part of the memory space utilized for the ASDF programs is consumed in the interfacing processes rather than in the methods by which the data is stored, used, or computed.

A. ALGORITHMS SELECTED

Because of the above stated objectives, a simple algorithm was selected for computation of time domain responses of digital filters. Once the user enters the desired pole and zero locations, a direct-form-two difference equation representation of the filter is generated [Ref. 5]. The unit sample sequence is used for the filter input, the difference equations are solved recursively in the discrete time domain and the unit sample response is recorded as the output. The unit step response is calculated by a similar method.

It is known that the direct-form-two filter implementation is very sensitive to parameter variations, and as such cannot be utilized in situations requiring extreme accuracy. The advantages associated with simplicity of understanding the algorithms, the ease of programming the general case, and the deemphasis of accuracy in outputs for marginally stable filters outweighs this disadvantage.

B. DEFINING THE DATA STRUCTURE

A detailed description of the important variables in the ASDF programs is given in Appendix B. There is, however, an overiding data structure common to all portions of the ASDF package which is important.

The array POLZRO holds the locations of the poles and zeros of the filter under analysis. In all cases, zeros are

stored in POLZRO(I,J) for values of I from one to ten. Pole locations are stored in POLZRO(I,J) for values of I from eleven to twenty. Only the top half plane pole of each complex pair is stored. There are five sections of POLZRO(I,J), i.e. values of J from one to five. POLZRO(I,1) is the real part of the rectangular representation of the ith root. POLZRO(I,2) is the imaginary part of the rectangular representation of the ith root. POLZRO(I,3) and POLZRO(I,4) are the magnitude and angle portions of the ith root location in polar form. POLZRO(I,5) is the root type designator. Root type designators are: one for real zeros, two for complex zeros, three for real poles, and four for complex poles.

The array IROOTS(I) holds the numbers of roots of each type of root already entered into the system. IROOTS(1) and IROOTS(3) hold the number of real zeros and poles, respectively. IROOTS(2) and IROOTS(4) hold the number of complex zero and pole pairs, respectively.

The variable IERROR is the error flag, which is set whenever the user generates an error. There are numerous possible errors. These include trying to delete roots which have not been entered, or adding roots which cause the maximum system order to be exceeded.

III. IMPLEMENTED SOLUTION

Having determined the general form of the solution it remained to generate the software implementation. There were five major areas to be programmed: the interface with the z-plane, the time response, the frequency response, the plotting of output curves, and the supervisory routines.

A. INTERFACE WITH THE Z-PLANE

Since the method by which the user modifies the pole and zero locations in the z-plane is of no theoretical interest, the discussion of this implementation will be brief. Extensive coverage of the subroutines employed and the definitions of the variables used is reserved for Appendix B.

The ASDF command which allows the user to make pole-zero modifications in the z-plane is POLZRO. This command initiates execution of a load module which is structured as depicted in figure 3-1. Each of the boxes shown represents a subroutine. Lines with arrow heads on both ends indicate that program control passes to the subroutine and then directly back to the calling routine with no intermediate branches. The main program, MAIN, initializes the required arrays, and tests to ensure that the user has generated a realizable causal filter.

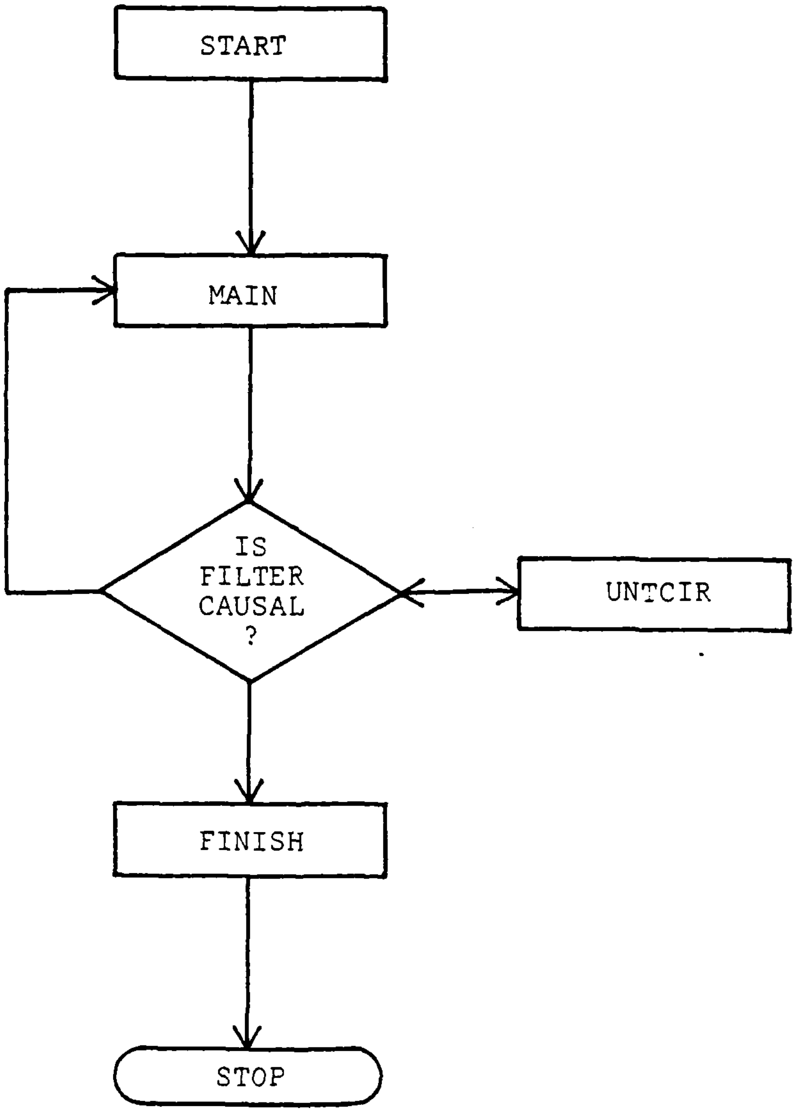


Figure 3-1 Flowchart Of POLZRO

The actual interaction with the z-plane is accomplished in subroutine UNTCIR. The structure of UNTCIR is diagrammed in figure 3-2. The large double ended arrow indicates that there are many subroutines of the form CRRMMM. The subroutines LETTER and BOXUC generate the unit circle and command array. PLOT-10 software [Ref. 10] statements determine the location of the user positioned cursor. This cursor position is adjusted with software tabs to one of twenty-six possible locations. Based on the X and Y location of the adjusted cursor position a specific subroutine is selected for execution. The flashing alphanumeric cursor is positioned in the selected command box using the adjusted cursor location as a reference. The subroutine then reads two alphanumeric characters typed by the user. The combination "space", "e" indicates that the user wishes the selected subroutine to be executed. The combination "space", "x" indicates that the user wishes to terminate the pole zero manipulation phase of the program.

All of the commands which change the status of the z-plane have names of the form CRRMMM where C is either "A" for add root(s) or "D" for delete root(s). The RR indicates root type: RZ - real zero, CZ - complex zero, RP - real pole, CP - complex pole. The MMM stands for the mode of data entry. IAG stands for interactive graphics i.e. the roots will be adjusted with the cursor. POL and RCT stand for polar and rectangular, respectively. The general

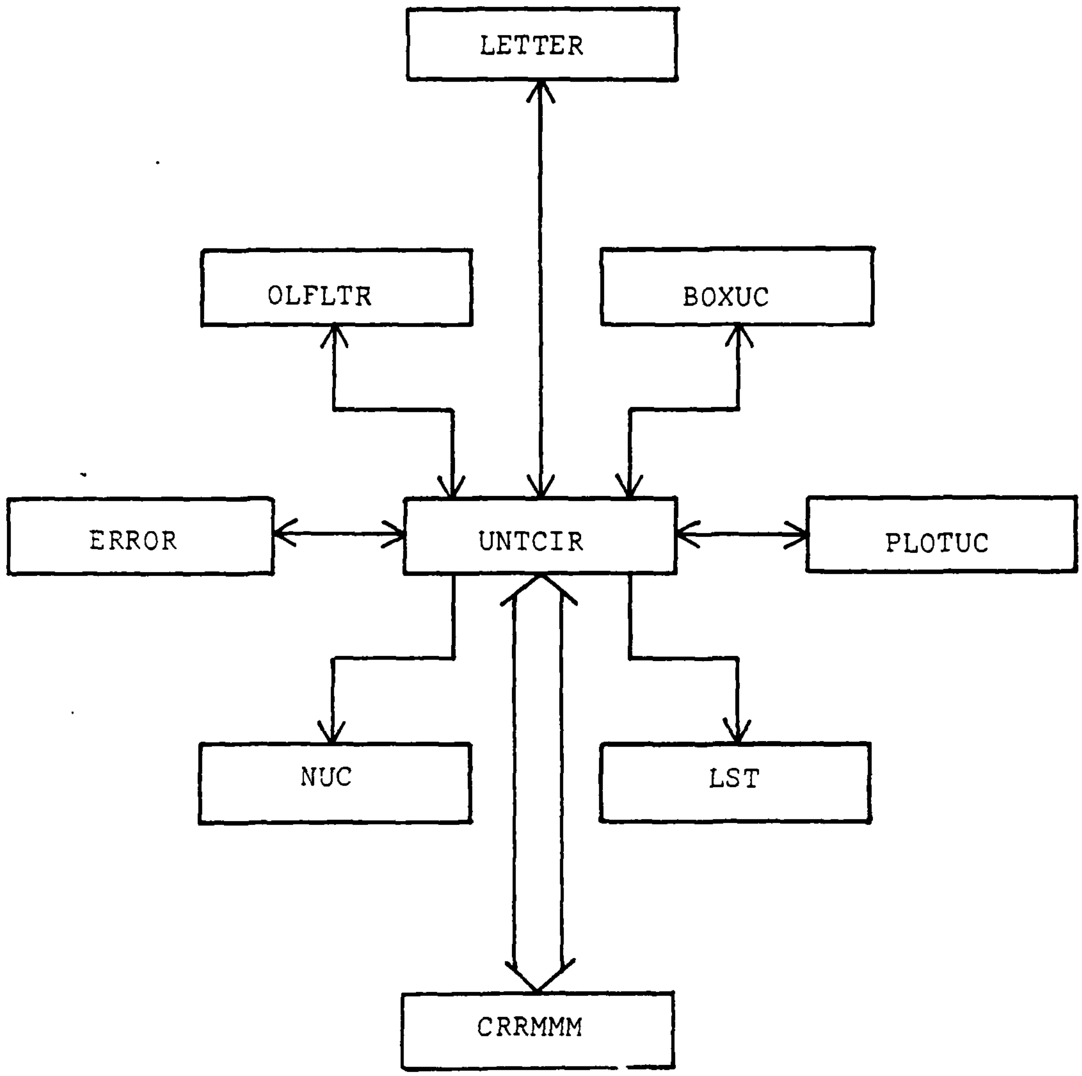


Figure 3-2 Structure Of Subroutine UNTCIR

structure of the CRRMMM subroutines is described in figure 3-3.

All CRRMMM subroutines follow the same general pattern. The POLZRO table is checked to determine whether or not the issued command is valid. Poles and zeros must be present before they can be deleted. There cannot be more than ten poles or ten zeros. The desired pole or zero location is next acquired. For subroutines adding roots[1] the entered location is that of the root(s) to be added. For subroutines deleting roots the entered location is any location near the root(s) to be deleted. Using subroutine LOCATE, the POLZRO table is searched for the root of the correct type which is nearest the specified location. This nearest root is then highlighted. In all cases, except when adding roots in the IAG mode, the user is asked whether or not the correct location has been determined. Once the root location is verified the root is added and plotted with subroutines STRRTS and PLOTRT. For subroutines deleting roots the POLZRO table is adjusted with subroutine UPDATE and the unit circle is redrawn.

The user may make an error in carrying out the above steps, i.e., by placing a pole outside the unit circle, by exceeding the maximum system order, or by entering a zero outside ten units from the origin. In each of these cases an

---

[1] Hereafter, poles and zeros will be refered to as roots.

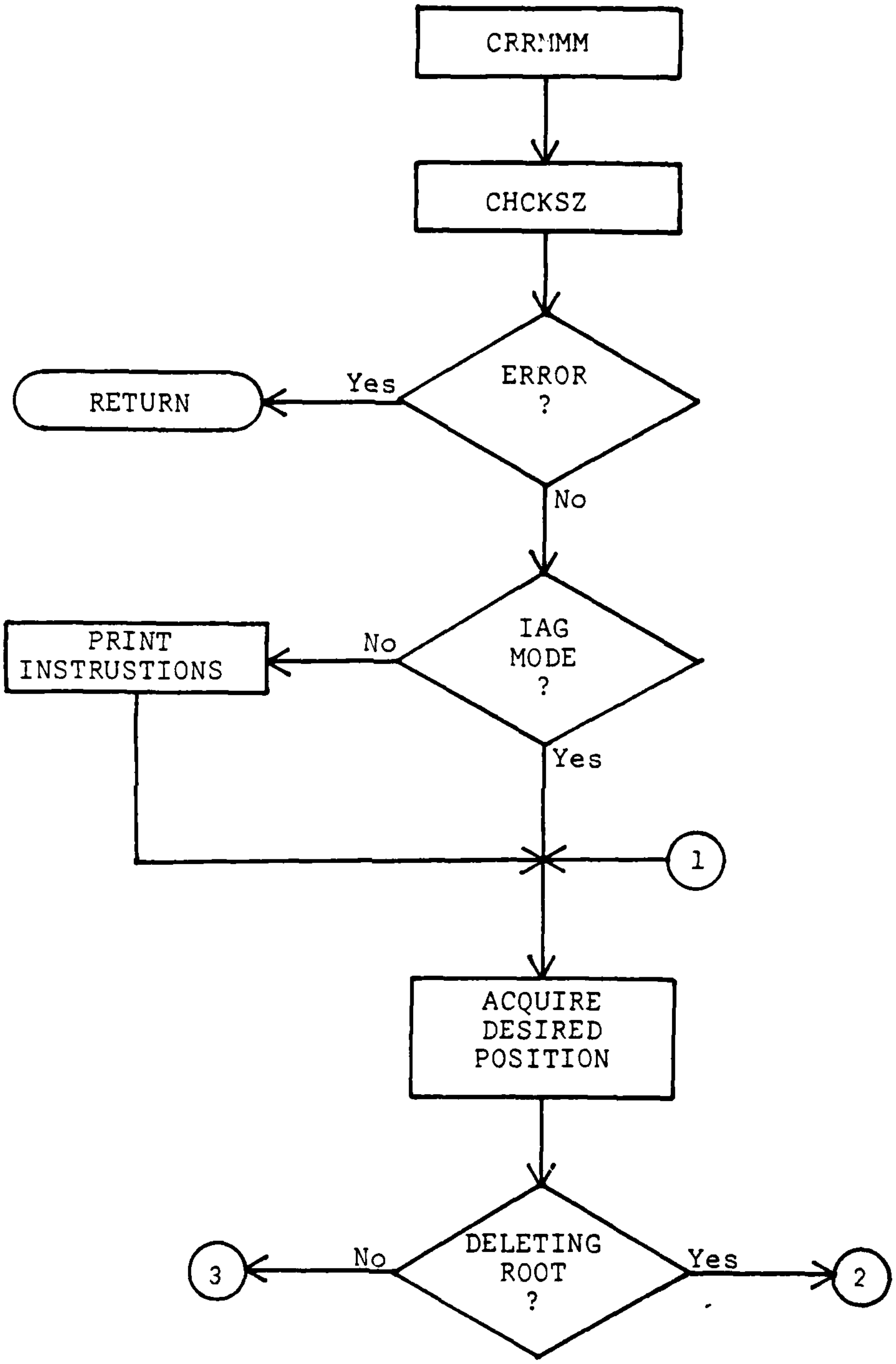


Figure 3-3a Structure of CRRMMM Subroutines

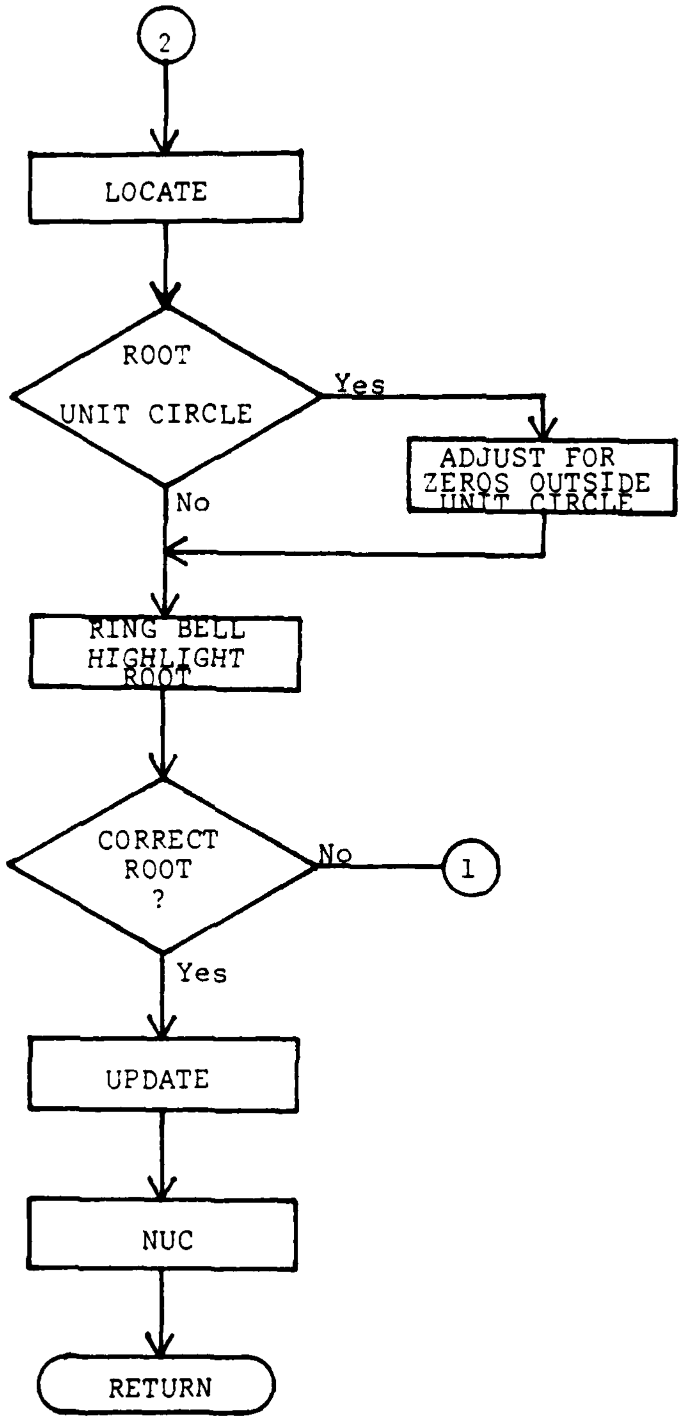


Figure 3-3b Structure Of CRRMMM Subroutines Continued

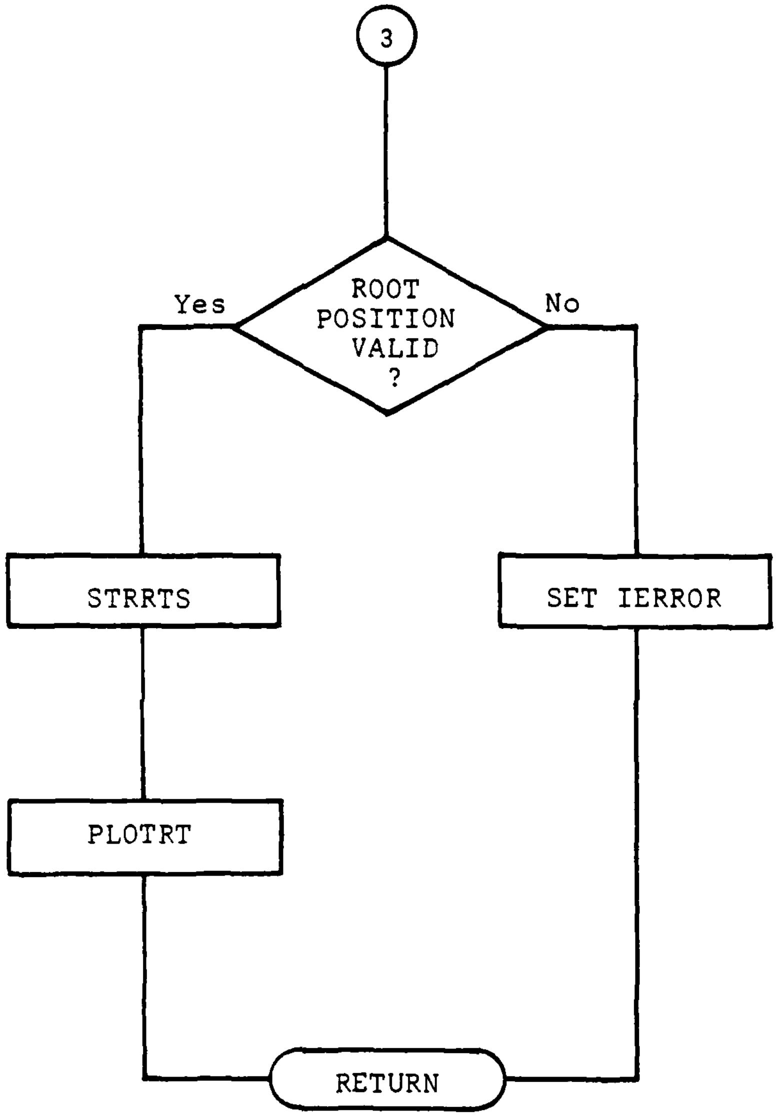


Figure 3-3c Structure Of CRRMMM Subroutines Continued

error flag is set. Subroutine ERROR handles the printing of messages and subroutine LST prints a listing of the current filter status before the program resumes.

B. TIME RESPONSE

The transfer function of a digital filter is defined in equation 3.1. The z_i, p_i, a_k, and b_r will be defined in a subsequent section. The variable A is an arbitrary constant assigned by the user and is called the filter gain.

$$H(z) = \frac{A \prod_{i=1}^{M} (z-z_i)}{\prod_{i=1}^{N} (z-p_i)} = \frac{\sum_{r=0}^{M} b_r z^{r-N}}{1 - \sum_{k=1}^{N} a_k z^{-k}} \quad (3.1)$$

Inherent in generating the time domain response with a direct-form-two filter implementation are the determination of the coefficients a_k and b_r in equation (3-1), and the computation of the unit sample and unit step sequences. Having determined the filter coeficients, the time domain response of the filter is determined by using the appropriate filter input sequence and recording the output of the filter for the desired number of points. This is accomplished by recursively solving a difference equation which characterizes the filter for 1024 time points. The standard direct-form-two filter implementation is shown in figure 3-4. The z^{-1} terms represent unit time delays. This

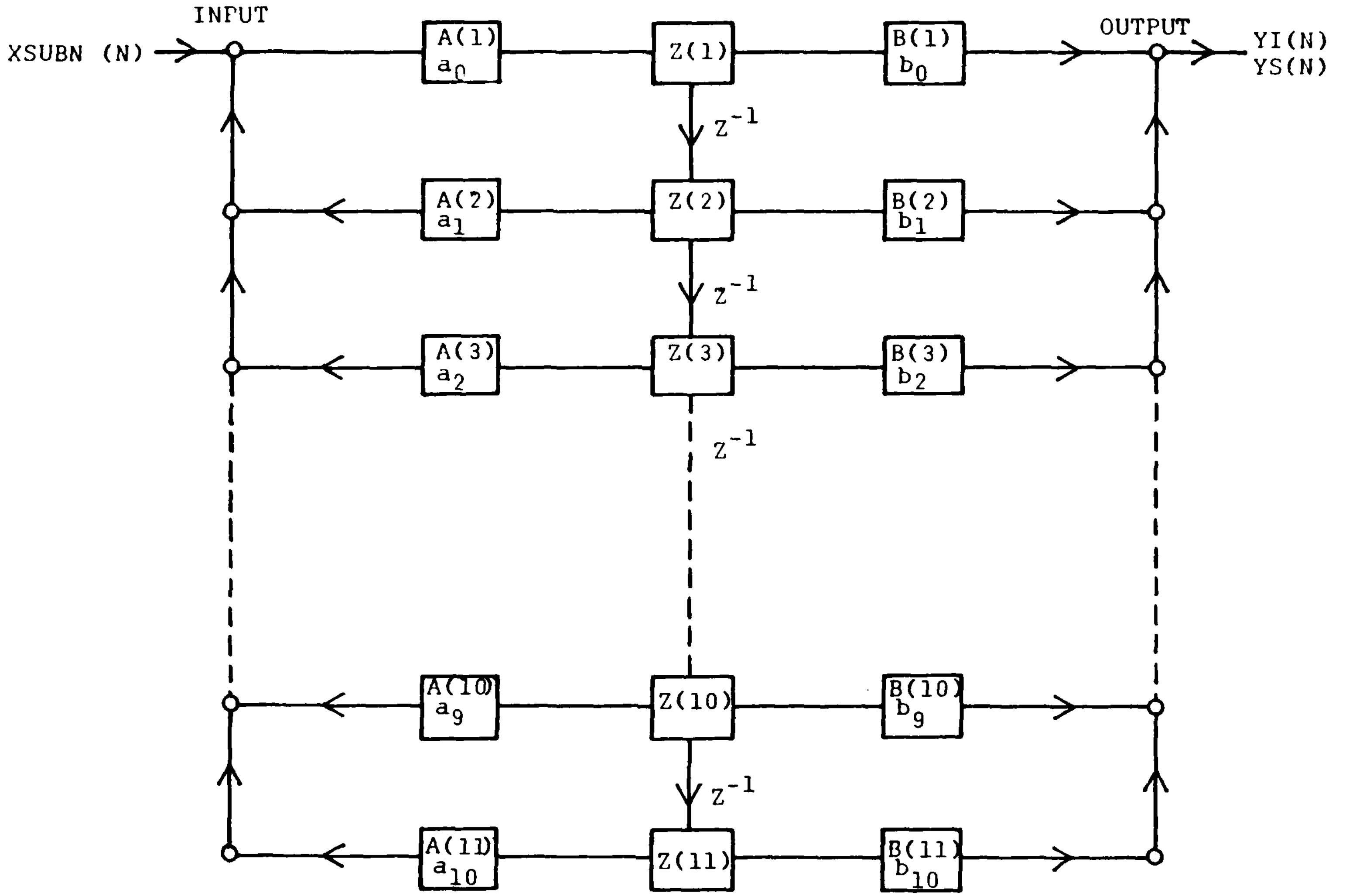


Figure 3-4 Direct Form Two Filter Implementation

figure corresponds to the difference equations:

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{r=0}^{N} b_r x(n-N+r) \qquad (3.2)$$

To compute the filter's time response, three operations are required: calculate the a_k and b_r coefficients, generate the input sequence x(k), k=0, 1, . . . 1024, solves the difference equations and stores the filter output values. There are four central subroutines which perform these operations: ASUBK, BSUBR, RSPNSE, and COEFF. Subroutines ABSUK, BSUBR, and COEFF generate the filter coefficients starting with the pole and zero locations specified. Subroutine RSPNSE solves the difference equations by coordinating the other three routines, generating the appropriate input sequence, and storing the output sequence values.

The ASUBK and BSUBR subroutines are identical in structure and purpose. ASUBK computes the degree of the denominator polynomial as two times the number of complex pole pairs plus the number of real poles. A complex vector is formed named POLES, which holds a listing of all the poles of the filter under analysis. The POLZRO array stores real poles in the form X+j0. Only the top half plane pole of each complex pole pair is stored, i.e., (X+jY), where X and Y are non-negative. The vector POLES has entries for all real poles and both poles of each complex pole pair.

Once the POLES vector has been generated the subroutine COEFF multiplies out the factors of the form $(z-p_i)$ to generate the coefficients of the denominator polynomial.

The algorithm for multiplying out the denominator polynomial is essentially the same as would be done by hand. multiplication of a series of literal factors of the form $(z-p_i)$. The vector POLES is passed to the COEFF subroutine as the vector V. Since the denominator factors are of the form $(z-p_i)$, the V vector is changed to hold the values $-p_i$. If the system is of order less than ten, the remaining positions of V are filled with zeros. The degree of the denominator polynomial is subsequently used to determine how many poles are actually located at the origin.

The vectors used in the subroutine COEFF hold the coefficients for the polynomial being generated. The literal powers of "z" are supressed. The vector H1 always contains the partial product of the K factors previously multiplied together. The vector H2 represents the old partial product times the "z" of the $(z-p_i)$ factor. Since the coefficient of "z" is always one, H2 is a left shifted version of H1. The vector H3 represents the partial product times the next pole location (p_i). This next $-p_i$ value is stored in V(1). The new H1 vector is formed by adding the H2 and H3 vectors. H1 now contains the product of the first K+1 factors. The values in the V vector are next shifted such that the next pole location to be multiplied is always

in the V(1) position. These four steps are repeated ten times to generate the coefficients of the complete polynomial in "z". The formula for the transfer function requires coefficients of the denominator polynomial in terms of z^{-1} so the order of the coefficients is reversed.

Figure 3-5 depicts several iterations of the algorithm used in the COEFF subroutine. Figure 3-5a shows the initial conditions for each of the vectors defined in the above paragraph. Figure 3-5c shows the values of the vectors after the first iteration. Figure 3-5f shows the vector values after the second iteration. The third and subsequent iterations follow this pattern.

In precisely the same way as discribed above, the a_k coefficients are generated with the ASUBK subroutine. The zero locations are stored in POLZRO exactly as are the poles. The generation of the vector ZEROS and the use of the subroutine COEFF parallels the BSUBR subroutine exactly.

The significant variables in subroutine RSPNSE are: XSUBN, INPUT, OUTPUT, FLTRGN, A, and B. The vectors A and B hold the already computed values of a_k and o_k. A(1) holds a_0; A(2) holds a_1 . . . A(11) holds a_{10}. The b_r are stored similarly in the B vector. XSUBN is the current value of the input sequence. INPUT is the value of the node so labeled in figure 3-4. OUTPUT is similarly noted on figure 3-4. FLTRGN is the value of the filter gain, and is assigned by the user. The vector Z holds the delayed INPUT values.

								V
H1	.	.	.	0	0	0	1	$-p_1$
H2	.	.	.	0	0	0	0	$-p_2$
H3	.	.	.	0	0	0	0	$-p_3$
								$-p_4$
								.
								.
								.

Figure 3-5a. Initial Conditions Of COEFF Vectors.

								V
H1	.	.	.	0	0	0	1	$-p_1$
H2	.	.	.	0	0	1	0	$-p_2$
H3	.	.	.	0	0	0	$-p_1$	$-p_3$
								$-p_4$
								.
								.
								.

Figure 3-5b. Values Of COEFF Vectors Before The End Of The First Iteration.

									V
H1	.	.	.	0	0	1	$-p_1$		$-p_2$
H2	.	.	.	0	0	1	0		$-p_3$
H3	.	.	.	0	0	0	$-p_1$		$-p_4$
									$-p_5$
									.
									.
									.

Figure 3-5c. Values Of COEFF Vectors starting The Second Iteration.

									V
H1	.	.	.	0	0	1	$-p_1$		$-p_2$
H2	.	.	.	0	1	$-p_1$	0		$-p_3$
H3	.	.	.	0	0	$-p_2$	$+p_1 p_2$		$-p_4$
									$-p_5$
									.
									.
									.

Figure 3-5d. Values Of COEFF Vectors before The End Of The Second Iteration.

H1	.	.	.	1	$-(p_1+p_2)$	$(p_1 p_2)$
H2	.	.	.	1	$-p_1$	0
H3	.	.	.	0	$-p_2$	$(p_1 p_2)$

V
$-p_3$
$-p_4$
$-p_5$
$-p_6$
.
.
.

Figure 3-5e. Values Of COEFF Vectors Starting The Third Iteration.

H1	.	.	1	$-(p_1+p_2)$	$(p_1 p_2)$
H2	.	1	$-(p_1+p_2)$	$(p_1 p_2)$	0
H3	.	0	$-p_3$	$+(p_1+p_2)p_3$	$-(p_1 p_2 p_3)$

V
$-p_3$
$-p_4$
$-p_5$
$-p_6$
.
.
.

Figure 3-5f. Values Of COEFF Vectors before The End Of The Third Iteration.

Z(1) is no delay; Z(2) is delayed one time unit; Z(3) is delayed two time units, etc.

The subroutine RSPNSE generates the appropriate input sequence. For the unit sample response the input sequence is 1, 0, 0, 0, ... For the unit step response the input sequence is 1, 1, 1, 1, ... The appropriate value of XSUBN is determined by the type of input sequence. The variable INPUT is then computed to be the sum of XSUBN and the a_k's times the delayed INPUT values. The filter output is the sum of the b_r's times the delayed INPUT values. These output values are then multiplied by the filter gain, FLTRGN, and stored in the appropriate array: YI for the unit sample response, YS for the unit step response.

C. FREQUENCY RESPONSE

For discrete systems, the frequency response is found by evaluating the system transfer function at points on the unit circle. Subroutine FRQNCY generates both the phase and the magnitude of the transfer function.

The important variables in the FRQNCY subroutine are: AA, BB, THETA, Z, NNUM, and NDEN. The variables AA and BB are the double precision values of the A and B vectors described in the section on time response. THETA is the angular displacement around the unit circle for which a value of the frequency response is calculated. Z is the

classical z-plane variable:

$$Z = \exp(j\ \text{THETA}) \qquad (3.3)$$

NNUM is the number of numerator terms. NDEN is the number of denominator terms.

The frequency response is computed for 1024 points around the unit circle. All points computed are for THETA greater than zero and less than π. The values of Z for THETA greater than $-\pi$ and less than zero are neither computed nor plotted since the magnitude of the transfer function is an even function of THETA and the phase response is an odd function of THETA.

The generation of the a_k and b_r coefficients has already been discussed in the section on time domain response. The evaluation of H(z) is straightforward. The value of THETA is incremented through 1024 steps from zero to π. For each value of THETA generated, the value of Z is computed. The value of H(z) is then computed in two parts. The numerator is the sum of NNUM terms of the form b_r times z^{-r} for r=0, 1, 2,. . . NNUM-1[1]. The denominator is one minus the sum of (NDEN-1) terms of the form a_k times z^{-k} for k=1, 2, 3, . . . NDEN-1[2]. Dividing numerator by denominator

---

[1] NNUM is equal to the degree of the numerator plus one.

[2] NDEN equals the degree of the denominator plus one.

yields H(z). The phase of H(z) is then determined using the principal values of the arctangent function; the results are stored in the vector YP. The magnitude of H(z) is computed, multiplied by FLTRGN, the filter gain, and subsequently stored in the vector YM.

D. OUTPUT PLOTS

The output plots generated by ASDF are generated by two different software programs each being different implementations of the same plotting requirements. One set of plots is generated using PLOT-10 software on the Tektronix 4012 terminal. Another set of plots utilizes the Versatec plotter and associated software. [Ref. 7] The PLOT-10 software is adaquately covered in reference 10. The Versatec software is explained in Technical Note-034 in the computer center.

The plots which are generated on both devices are essentially the same. There are five plots generated on the 4012 terminal: unit sample response, unit step response, phase response, magnitude of the transfer function, and magnitude of the transfer function in decibels. Both of the time domain plots are terminated when the output value has stabilized to within one percent of the maximum generated system output. All 1024 points of the phase and magnitude plots are plotted.

The form of the Versatec output is similar to that of the 4012. One additional plot is included. It shows the unit circle with poles and zeros of the filter being analyzed. Plotting is terminated when the filter output stabilizes to within one percent of the maximum system output generated.

On both devices, the values of unit sample response are stored in a vector YI and plotted with subroutine PLTIMP. The unit step response values are stored in YS, and plotted with subroutine PLTSTP. The phase response is stored in vector YP. The magnitude of the transfer function is stored in vector YM. The magnitude of the transfer function in decibels is stored in YMDB. The three vectors YP, YM, and YMDB are all plotted with subroutine PLTTRF. The Versatec plot of the unit circle is generated with subroutine UNTCR. The 4012 plot of the unit circle and command array are generated with the subroutine UNTCIR.

E. SUPERVISORY ROUTINES

There are three supervisory routines used with the ASDF programs. These supervisory routines make the loading of ASDF transparent to the user. There is a substantial amount of file manipulation required to store and load the ASDF programs. A knowledge of the hardware and software available for these purposes is not particularly useful to the user. For these reasons, the supervisory routines are discussed only briefly. Two of these routines are based on

CP/CMS and IBM EXEC environment; the third is written in the IBM-360 Job Control Language.

The JCL program moves the ASDF program from its permanent storage location on DISK02 to the CP/CMS disks OSX001. The two EXEC routines generate an appropriate CP/CMS work space, and manipulate the ASDF text file into a usable form.

The EXEC program ASDF191 resides on the user 191-P-disk. This routine automatically requests and formats ten cylinders of temporary disk space. The ASDF TEXT file is moved from the OSX001 disk to the newly created temporary disk space 192-T. Once in place, The ASDF TEXT file name is changed to ASDF EXEC to generate the next file with the correct file type. ASDF EXEC is then edited to remove the first set of data cards which hold the ASDF192 EXEC program. The 192-T-disk is released and the user is requested to type two commands to the system. The "login 192 P" command makes the 192 disk the P-disk. The "EXEC ASDF192" command begins the execution of the second EXEC file.

The general ASDF192 EXEC structure is depicted in figure 3-6. ASDF192 EXEC is created by ASDF191 EXEC and resides on disk 192. In a manner similar to ASDF191, ASDF192 splits the ASDF TEXT file, which contains all of the component parts of the ASDF program, into a series of smaller files. Once all files have been separated, titles are altered appropriately, text files are loaded into core, and load

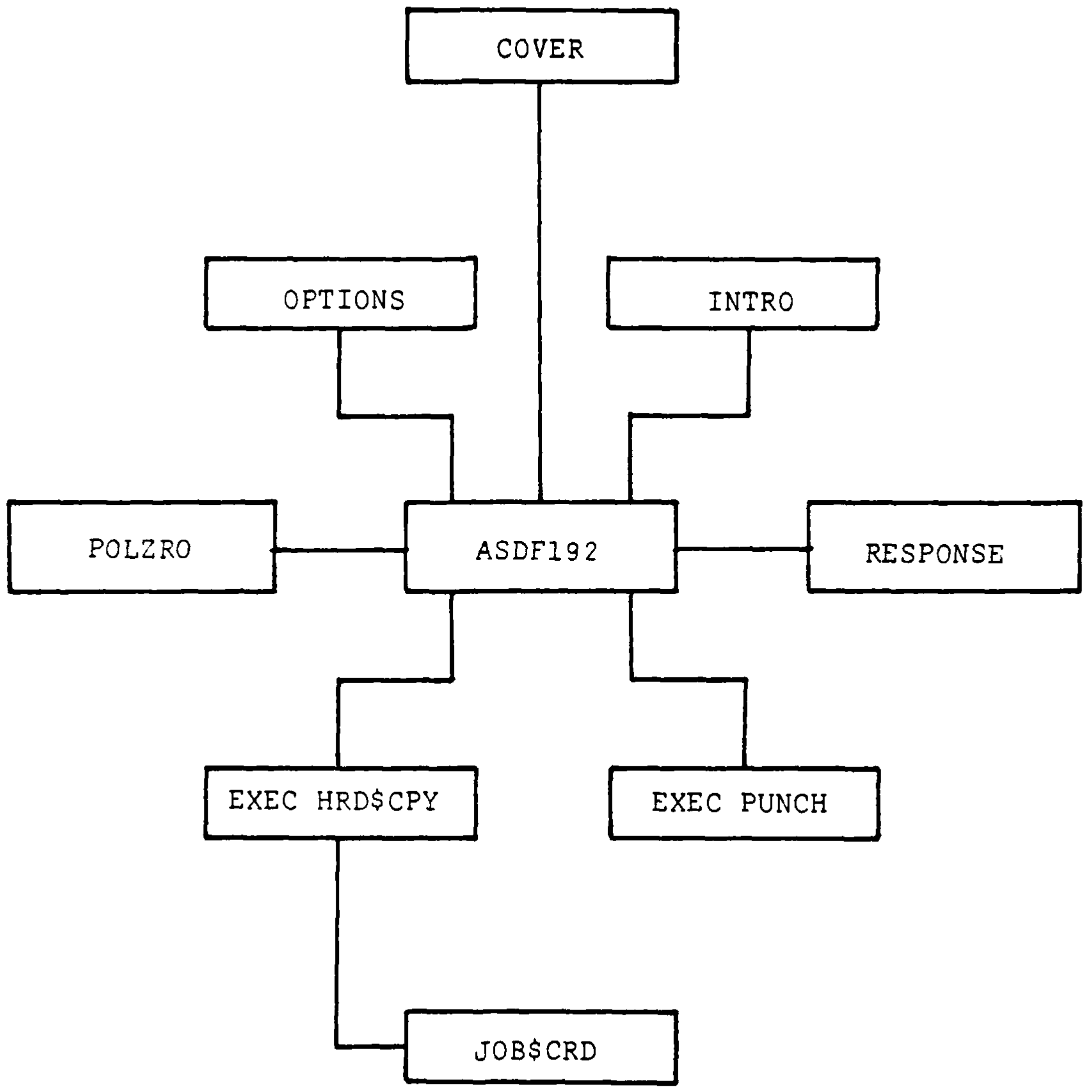


Figure 3-6 Structure Of ASDF192 EXEC

modules are generated. ASDF192 then reverts to an infinite loop, reading and implementing the desired user command. There are six user commands available to the user: POLZRO, RESPONSE, EXEC PUNCH, EXEC HRD$CPY, OPTIONS, and INTRO. The applicability of these commands is briefly discussed in chapter four. This brief description is also presented by executing the OPTIONS command once ASDF has been loaded.

IV. STRUCTURE OF ASDF COMMANDS

The ASDF program generates the time and frequency responses of realizable digital filters of order ten or less. Chapter three describes the algorithms used for these computations. The procedure for executing the ASDF program generally consists of two distinct steps. The first step is to enter the poles and zeros of the filter to be analyzed. The second step is to perform the calculations required to generate and plot the desired filter responses. There are six commands which may be executed within the ASDF environment providing the user with the ability to control the form of the output, to enter the poles and zeros of the filter, and to clarify most questions pertaining to the meaning and use of the general commands. The six fundamental commands available to the user are: OPTIONS, INTRO, POLZRO, RESPONSE, EXEC PUNCH, and EXEC HRDSCPY. Only the POLZRO command has a subordinate command structure. All of these commands are issued by typing the command on the terminal keyboard in response to: *ASDF-RE.

The OPTIONS command provides a listing of all commands available to the user. Each command is accompanied by a brief description of its intended use.

The INTRO command writes on the screen six pages of information pertaining to the POLZRO command substructure.

INTRO includes a general description of the data input modes, and a description of several input situations with the appropriate executable commands.

The POLZRO command allows the user to modify or create a filter in the z-plane. All adjustment of poles and zeros must take place with this command. By issuing this command the user may add and delete poles and/or zeros as well as adjust the filter gain constant. The commands which are issued within the POLZRO environment are described in a later section of this chapter.

The RESPONSE command displays the characteristics of the filter being created by the user. Five displays are presented: unit sample response, unit step response, phase of the transfer function, magnitude of the transfer function, and the magnitude of the transfer function in decibels.

The EXEC PUNCH command allows the user to punch a card deck of the filter pole and zero locations on the offline punch. This punched output can be picked up in room 1-140 of the computer center. The cards punched by this command may be subsequently reentered into the system so that this particular filter can be referenced or modified at a later date.

When the user desires to enter a filter in this way, the punched cards are presented to one of the computer operators over the counter in room 1-140. The cards are punched in

the correct format for direct submission into the CP/CMS card reader. The operator subsequently reads this deck into the user's virtual card reader. As the user signs onto a terminal and gains access to CP/CMS, the computer includes an additional message in the standard login typeout showing that the user has a virtual reader file. This message is of the form:

FILES:- 01 RDR, NO PRT, NO PUN

When the user wishes to access this old filter data he must execute the CP/CMS command OFFLINE READ * *. The OFFLINE READ command will read the punched card filter data from the user's virtual card reader, and generate a file on the user's disk space entitled FILE FT01F001. Any files with this name present before the OFFLINE READ command is issued will be over-written by the new file. If the user does not wish to lose an older FILE FT01F001, he should use the CP/CMS ALTER command to change the name of the old file that he wishes to protect. Once the user has successfully loaded the ASDF program, the ASDF signature i.e., *ASDF-RE, is presented. The OFFLINE READ command may be issued any time that this signature appears.

The EXEC HRDSCPY command is available only when the SUBMIT function of CP/CMS is functioning. When the SUBMIT command is not functional a message is sent to the user during the login procedure which states that SUBMIT is not available. the output from this command is ultimately

available in the computer center, and consists of two parts. The first part of the output is the numerical listing of the values of the responses computed by the ASDF program. The EXEC HRD$CPY command is the only command which will allow the user access to these numerical values. The second part of the output is a series of Versatec plots, each similar to the screen presentations of the RESPONSE command, with one additional plot of the unit circle and pole/zero locations. When the EXEC HRD$CPY command is executed the ASDF program will ask several questions, using the responses to generate a JOB card. The response to the "ENTER DESIRED JOB NAME" is the title which appears on the final output. ASDF then submits the filter characteristics to the operating system for processing. The hard copy output will be found in the standard output boxes.

A. SUMMARY OF DATA INPUT MODES

There are three modes through which the poles and zeros may be entered into the z-plane: 1) interactive graphics, 2) rectangular, and 3) polar. Each of these modes has particular attributes and limitations.

1. Interactive Graphics (IAG)

This mode was designed to enter poles and zeros rapidly, in cases where accuracy is not of prime importance. The interactive graphics mode is characterized by the appearance of the graphics cursor (the intersection of two

lines which trace across the terminal screen). The position of the cursor is controlled by the two thumb wheels located to the right of the interactive ASCII keyboard on the Tektronix 4012 terminal. The coverage of the interactive graphics cursor is limited to values of X and Y greater than -1.1 and less than +1.1. Zeros outside this area must be entered with either the polar or rectangular commands. Careful adjustment of the cursor can usually locate poles or zeros within .002 units of the desired location. Poles must be located within the unit circle.

2. Rectangular

This mode was designed to enter from the terminal keyboard, poles and zeros which are known in rectangular coordinates. It is slower than the interactive graphics mode because there is a significant amount of screen rewriting required. The rectangular mode allows much greater accuracy in the location of poles and zeros. Poles may not be entered on or outside the unit circle. Zeros must be located within ten units of the z-plane origin.

3. Polar

This mode was designed to enter from the terminal keyboard, poles and zeros which are known in polar coordinates. This mode is also slower than the interactive graphics mode. It allows poles and zeros to be entered to any permissible position in the z-plane. Poles must be entered within the unit circle. Zeros must be entered within

ten units of the z-plane origin. Each polar command has a description of the constraints on the data to be entered automatically written on the screen. This mode is particularily appropriate when constructing all-pass linear phase filters.

B. THE POLZRO COMMAND SUBSTRUCTURE

In an attempt to maximize execution speed, a minimum of information is written on the screen, thus reducing the rewrite time after execution of each command. When program execution begins, the user is presented with the unit circle, and a listing of commands on the top left side of the screen. The commands are abbreviated in the following manner. The general command type or mode is specified across the top of the array as INTERACTIVE, RECT, and POLAR (see figure 4-1). Each of these modes conforms to the above description. Down the left side of the array are the commands of the form CRR, where C is the change to be made, A - for add a root, D - for delete a root. RR is for the type of root to be acted upon by the command: RZ - real zero, CZ - complex zero, RP - real pole, and CP - complex pole. As an example, the box in the center column aligned with the row titled DCZ stands for the command "Delete a Complex Zero - Rectangular Coordinates. There are two unique commands listed as NUC, and LST. The New Unit Circle (NUC) command simply redraws the unit circle and command

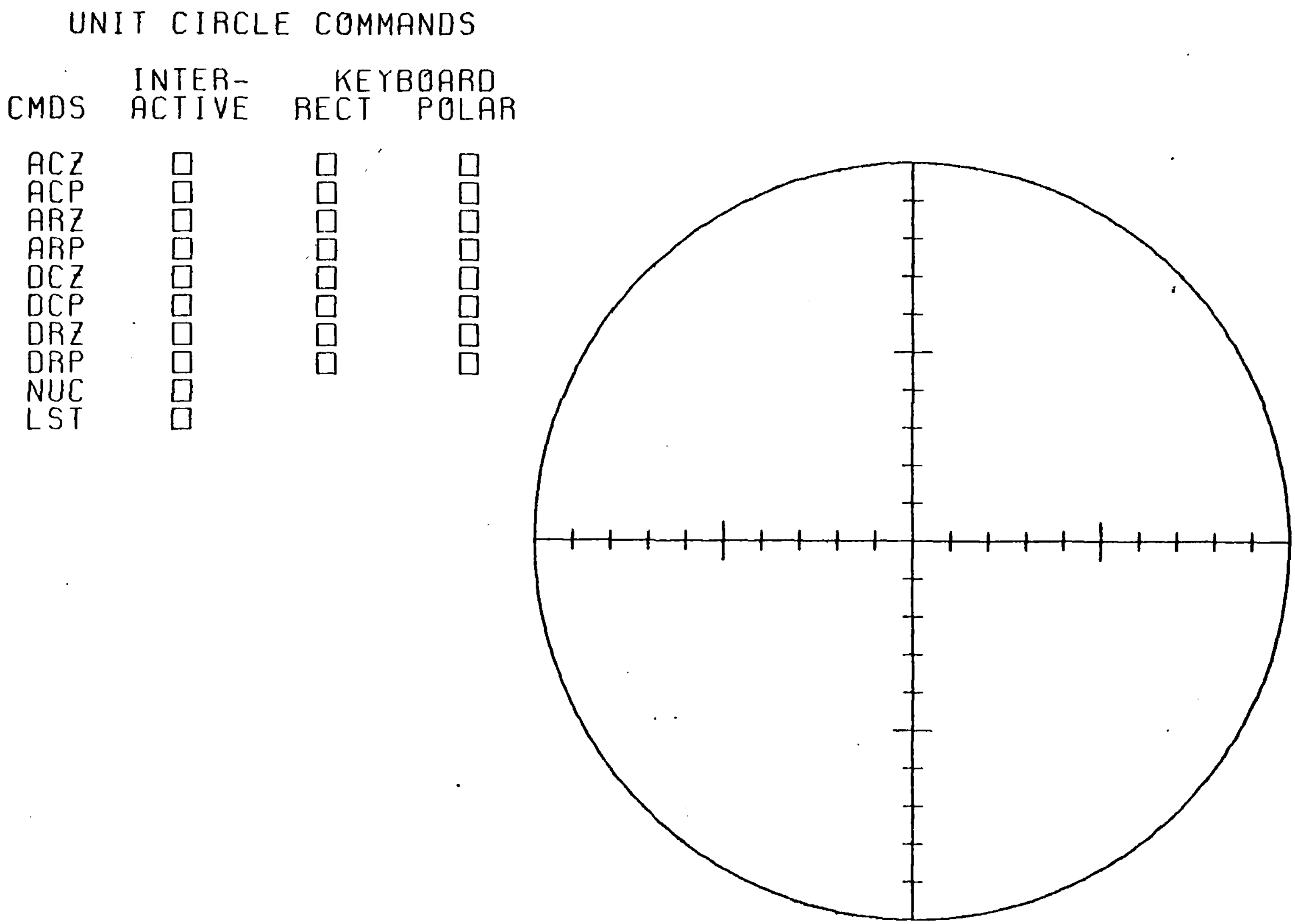


Figure 4-1 Basic Display Format

array. This is appropriate when either too many entries have been made to the terminal, and the command array is no longer clear, or when noise in the line from the computer causes a deformed array or unit circle to be drawn. The LST command provides the user with a listing of the current status of the filter being analyzed, in a self explanatory format.

In all cases when the unit circle and command array just described appear on the screen, an audible tone or bell is heard. This bell indicates to the user that the cursor is to be positioned within one of the boxes in the command array. In order to execute a command, the cursor is positioned in the box associated with the desired command. A single keyboard character (any character but the "return" key is acceptable) is then typed. The cursor then disappears. The system presents a carat pointing to the command that the computer is about to execute; the flashing cursor blinks within the box associated with the command which is about to be executed. If the box so indicated is the command that was selected, the user types a "space", an "e" (for execute) and then "carriage return". In cases where a rectangular or polar mode command is to be executed, the screen is then erased, and new information is printed. In cases where an interactive graphics command is to be executed, only the cursor reappears. This second appearance of the cursor is not accompanied by a bell or audible tone.

This indicates that the cursor is to be positioned within the unit circle. Once so positioned, a single keyboard character is typed (not "return"), and, as appropriate, the unit circle is redrawn to reflect the change in pole or zero locations. After this update of the z-plane, the circle and command array are redrawn with the cursor accompanied by the audible tone.

There are only two exceptions to this scenario. First, if the flashing cursor does not flash in the correct box, the wrong command will be executed. If an incorrect box contains the flashing cursor, the user should not type the "space", "e", sequence. Instead, he should type two other characters (e.g., "space", "space") and then "return." The cursor returns along with the audible tone, calling for a readjustment of the cursor, closer to the desired command box. The second exception occurs when the placement of all of the poles and zeros is completed. In this case, after aligning the cursor with any box, the sequence "space", "x" (for exit), "return" should be typed. This will terminate the input, or modification, stage of the program and present the user with a series of options for the next stage of filter processing. At this point a display appears which explains the options available to the user.

C. SUMMARY OF POLZRO SUBCOMMANDS

When the cursor appears along with an audible tone, the user is to position the cursor within the command array, and type a keyboard character (not "return"). When the flashing cursor appears within a box, he types ("space", "e", "return") to execute the specified command. "Space" and any other character but "x" will return the user to the command table for cursor readjustment. To terminate the input, or modification mode, the cursor is positioned over any box, and once the flashing cursor enters the box, the characters "space", "x", "return" are typed.

V. LOADING THE ASDF PACKAGE

The procedure for using the ASDF program consists of either six or seven steps depending on whether or not the ASDF package is resident on OSX001. Usually the program sections of ASDF are stored on the T2314 disk pack OSX001. Each Monday morning the OSX001 disks are scratched so that the ASDF program must be moved from the permanent storage location on DISK02, to the temporary storage location on OSX001. The user may simply sign onto a 4012 terminal, and execute three commands to load ASDF into his virtual machine. Once this loading process has been successfully completed, the commands previously discussed are at the user's disposal for filter analysis.

A. LOGIN PROCEDURE

With one minor exception, the standard login procedures (specified in Section 4.4.3 of the CP/CMS User's Manual) [Ref. 4] are applicable when entering the CP/CMS system to use ASDF. Due to the size of the ASDF programs the user must request a 400k virtual machine when loging onto CP/CMS. The virtual machine is requested as follows:

login XXXXTNN 400k

where XXXX is the user's computer center number, T is the type of user i.e. G for general user, and P for private

user, and NN is the two digit terminal number. The 400k request in the LOGIN is absolutely mandatory. Without this request the user will be able to load the ASDF program, however, when attempting to execute the RESPONSE command, an error: "INPUT COMMAND ERROR" will result. Should this error occur when the command is corrctly spelled the user has no alternative but to logoff and log back on to the system requesting the correct size virtual machine. The remainder of the login procedure is identical to the examples in the CP/CMS user's manual.

B. TESTING OSX001 FOR ASDF PROGRAMS

The second step in loading the ASDF programs is to determine whether or not a copy of the programs exists on OSX001. The ASDF programs are permanently stored on the operating system DISK02. The first time that the programs are used each week, the user must move them from DISK02 to the disk OSX001. To determine whether or not OSX001 has the ASDF package, issue the command:

```
TOCP ASDF191 EXEC P1
```

The system will respond:

```
EXECUTION BEGINS...
237 = OSX001
ENTER DSNAME OR '/*'
```

The correct user response is:

```
F0718.asdf191
```

At this time one of two responses will appear. The particular response received is dependent on whether or not ASDF resides on OSX001.

1. <u>ASDF is on OSX001</u>

When the ASDF programs are located on OSX001, the system response is:

EXTENTS EXHAUSTED, EOF SIMULATED.

4 BLOCKS 63 RECORDS READ

This message indicates that file ASDF191 has satisfactorially been transferred to the user's virtual machine and the user is ready to continue with the subsequent loading steps.

2. <u>ASDF is not on OSX001</u>

When the ASDF programs are not on OSX001, the system response will be:

DATA SET NOT FOUND (OS/360 S213)

ENTER DSNAME OR '/*'

This response indicates that the user must move the ASDF programs to OSX001 before the loading procedure may be resumed. The simplest method for accomplishing this transfer is to generate a file on CP/CMS which exactly duplicates the Job Control Language program which follows.

To generate the file which transfers ASDF, logon to CP/CMS. Once the system's responses have been entered and the system is ready, issue the command:

EDIT ASDF JCL

Then type in, or read in[1], the program which follows.

```
// Standard JOB Card (User's Manual Sec. 3.3.2.1)
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=F0718.ASDF,VOL=SER=DISK02,UNIT=3330,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),DISP=(OLD,KEEP)
//SYSUT2 DD DSN=F0718.ASDF,VOL=SER=OSX001,UNIT=2314,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),DISP=(NEW,KEEP),
//   SPACE=(TRK,(40,1),RLSE)
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=F0718.ASDF191,VOL=SER=DISK02,UNIT=3330,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(OLD,KEEP)
//SYSUT2 DD DSN=F0718.ASDF191,VOL=SER=OSX001,UNIT=2314,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=-800),DISP=NEW,KEEP),
//   SPACE=(800,(3,1),RLSE)
//
```

Once this program has been correctly read in to the user's disk, it must be submitted to the operating system using the standard SUBMIT command:

SUBMIT ASDF JCL P1 XXXX

XXXX is the user's computer center number. If the SUBMIT function is working the following messages will be returned:

** CARDS XFERED TO 0098P **

FNAME FTYPE P1 SENT FOR OS BATCH JOB

After a short delay the system will send a second message:

FROM 0098P33 : JOB SENT TO OS WITH 21 CARDS WRITTEN (Time)

The user should then revert back to section B above which

---

[1] A deck containing this program is available from Professor Kirk, Sp 302.

discusses the TOCP command. After giving the system a period of time to execute the JCL job which was just submitted, execution of the TOCP command will indicate that the ASDF files are present. If the operating system has not completed executing the transfer, the TOCP request will respond as in paragraph two above.

B. LOADING ASDF

Once ASDF resides on disk OSX001, the remaining required steps are simple. The user issues the command:

EXEC ASDF191

The system will type a series of lines onto the terminal. At the conclusion of these typewritten lines the user is prompted to issue the next two commands:

LOGIN 192 P

The system responds:

192 REPLACES P (191)

The next required entry is:

EXEC ASDF192

These two commands will cause a very long list of CP/CMS commands to be typed on the screen. Each of these CP/CMS commands is automatically executed by the operating system in the process of loading the ASDF programs. There are no furthrer commands required from the user to complete the loading process.

When the entire ASDF program package is loaded, the user will be presented with a title page and the instruction: "HIT SPACE AND RETURN TO CONTINUE." Once the user has typed "space" and "return" the display will write out a listing of the commands at his disposal. These commands have already been discussed in chapter four.

VI. USING ASDF

Each of the following sections constitutes an example for one of the three input modes. Each example uses only one input mode. In general, these modes may be combined during the generation of any particular filter. The mode selected should be the one most convenient to the user.

A. EXAMPLE ONE - INTERACTIVE GRAPHICS

This first example uses the ASDF program in the interactive graphics mode. The objective is to observe the responses of a digital filter which has two real zeros, one at the origin and another at +0.8, two complex poles located at (0.8 +/- j0.2), and the filter gain is 1.000. The interactive graphics mode is used since the exact locations of the poles and zeros are not critical. For the purposes of this example, it is assumed that the user has already followed the login procedure described in the above paragraphs, and has loaded the ASDF program. The user has already seen the title page, and the two pages of command descriptions. The screen is clear except for the top left corner which shows:

*ASDF-RE

This group of letters is the signature for the ASDF program

and indicates that the program is ready to execute the user's next command.

Since we must first enter the poles and zeros into the z-plane the appropriate command to type is:

POLZRO

The ASDF program will then draw the screen as shown in figure 6-1, and generate an audible tone. Initially the cursor will be located at some arbitrary position or perhaps may not appear on the screen at all. Movement of the thumb wheels moves the cursor position. Since we wish to execute in the interactive graphics mode, to add a real zero, we move the cursor (intersection of the two lines tracing across the screen) into the box in the left column labeled ARZ, (Add a Real Zero) figure 6-2. We then type any keyboard character except "return".[1] The trace cursor will disappear, and the flashing alphanumeric cursor will appear. The program will print a carat pointing to the box selected by the user. The flashing cursor will move into the box selected.

The user then types the command "space", "e", "return". The flashing cursor disappears, and the trace cursor reappears. The absence of an audible tone at this time indicates that the trace cursor is to be positioned within the unit circle.

---

[1]The 4012 terminal in the computer center requires a "control-s" after every response.

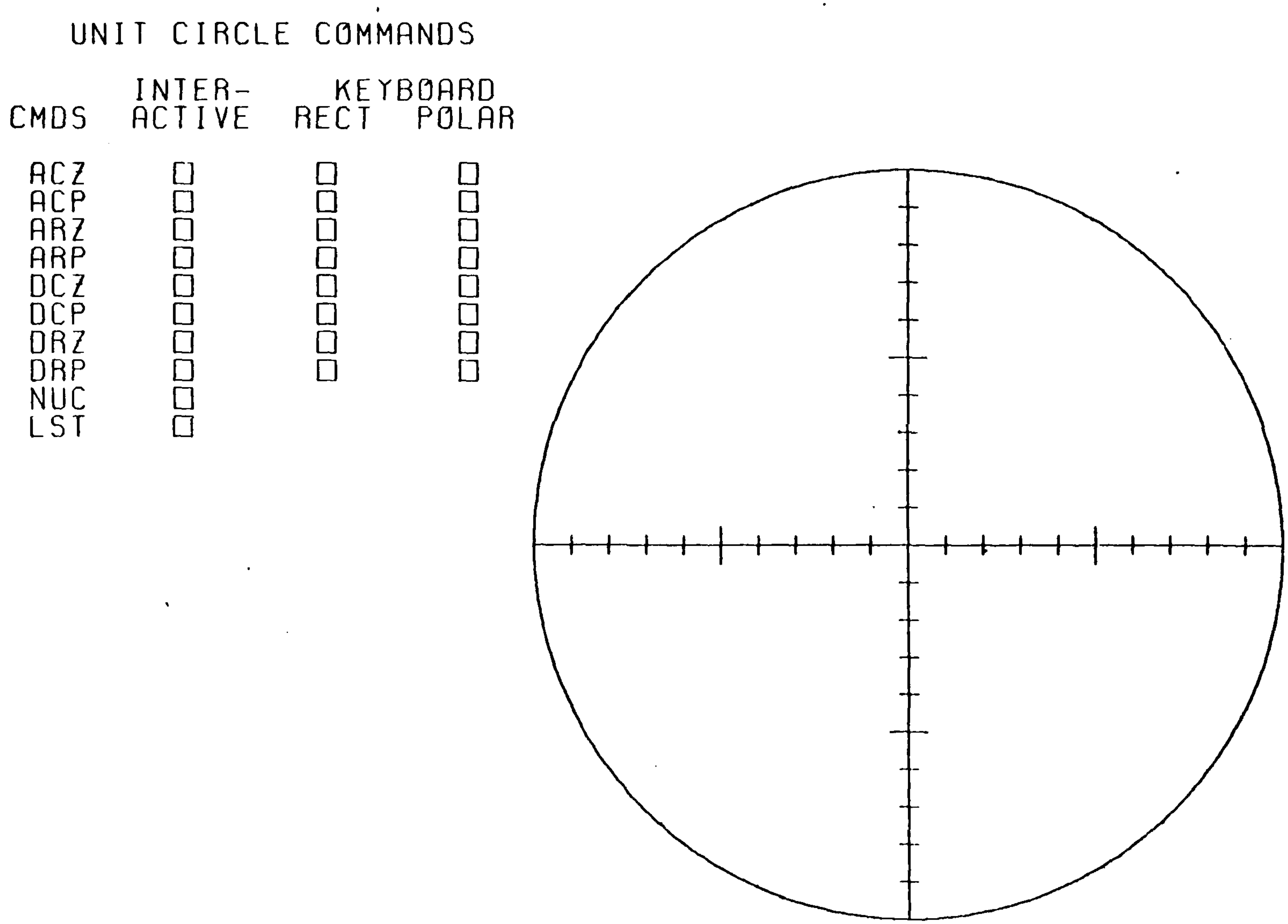


Figure 6-1 Initial Screen Representation After Issuing The Command POLZRO

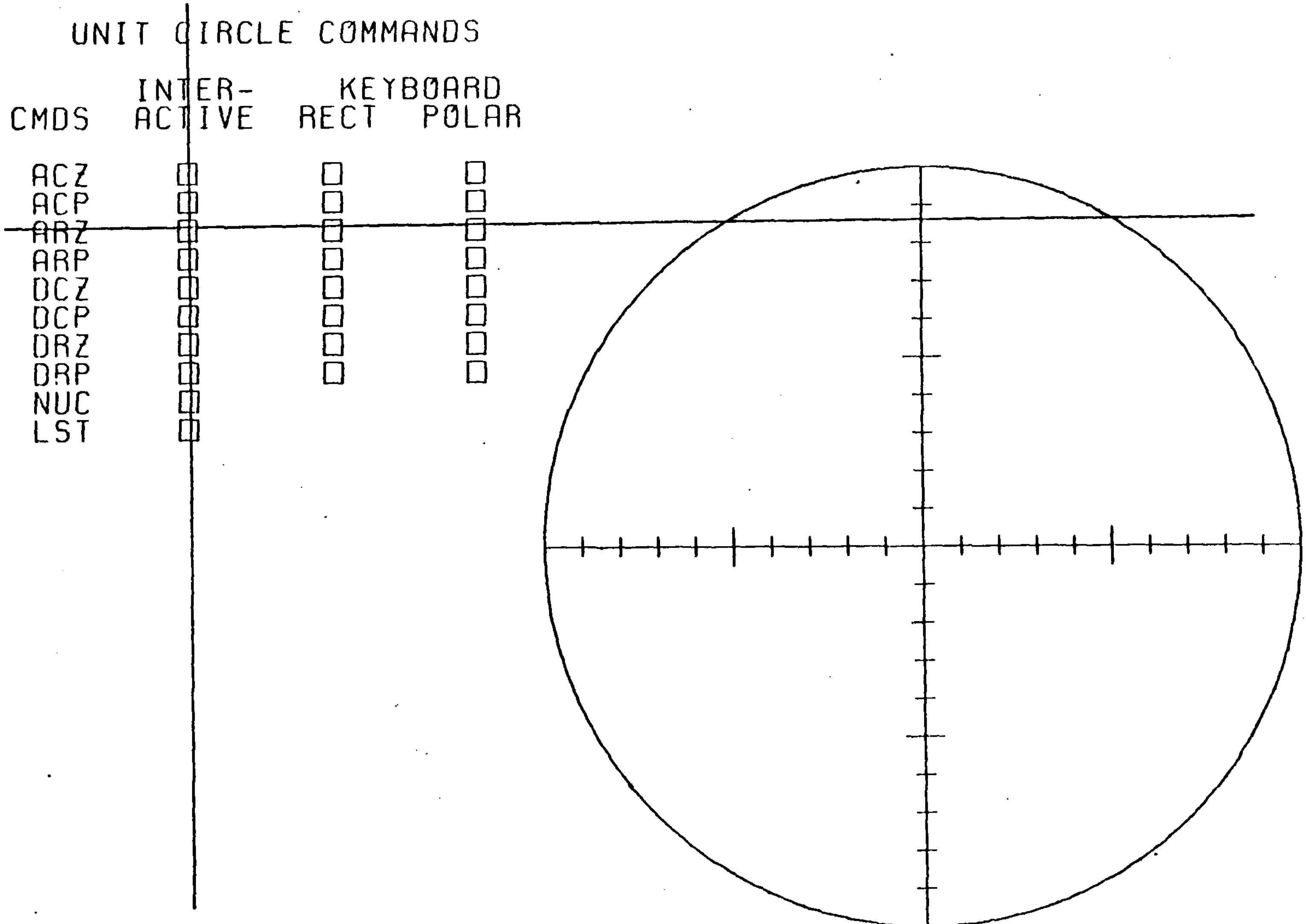


Figure 6-2 Trace Cursor Position For Executing The Command "Add A Real Zero"-Interactive Graphics Mode

The trace cursor is adjusted to the position of the desired real zero, (the origin in this case), as shown in figure 6-3. Any keyboard character except "return" is then typed. The ASDF program plots the zero at the origin, and returns the trace cursor with the audible tone. The audible tone reminds the user that the trace cursor is to be positioned in the command array. Since another real zero is to be entered, the cursor is positioned as in figure 6-1b again, and the same steps are followed. When the trace cursor reappears without the audible tone, position it within the unit circle as near as possible to the point (0.8 +/- j0.0) as shown in figure 6-4. (To enter the zero precisely at 0.8 the rectangular or polar mode would be used.) The steps for entering the complex pole pair exactly parallel those for the previous entries, and are depicted in figures 6-5, and 6-6.

Once all the poles and zeros have been entered, the trace cursor may be aligned with any command box, and a character typed (not "return"). Once the alphanumeric cursor enters that box, the user types "space", "x", "return". The screen is cleared and the user is asked if he wishes to change the default filter gain. Since the default value is 1.0, the user responds "n", the ASDF program clears and responds with: *ASDF-RE

The next step in the normal sequence is for the user to look at the filter characteristics. To display these

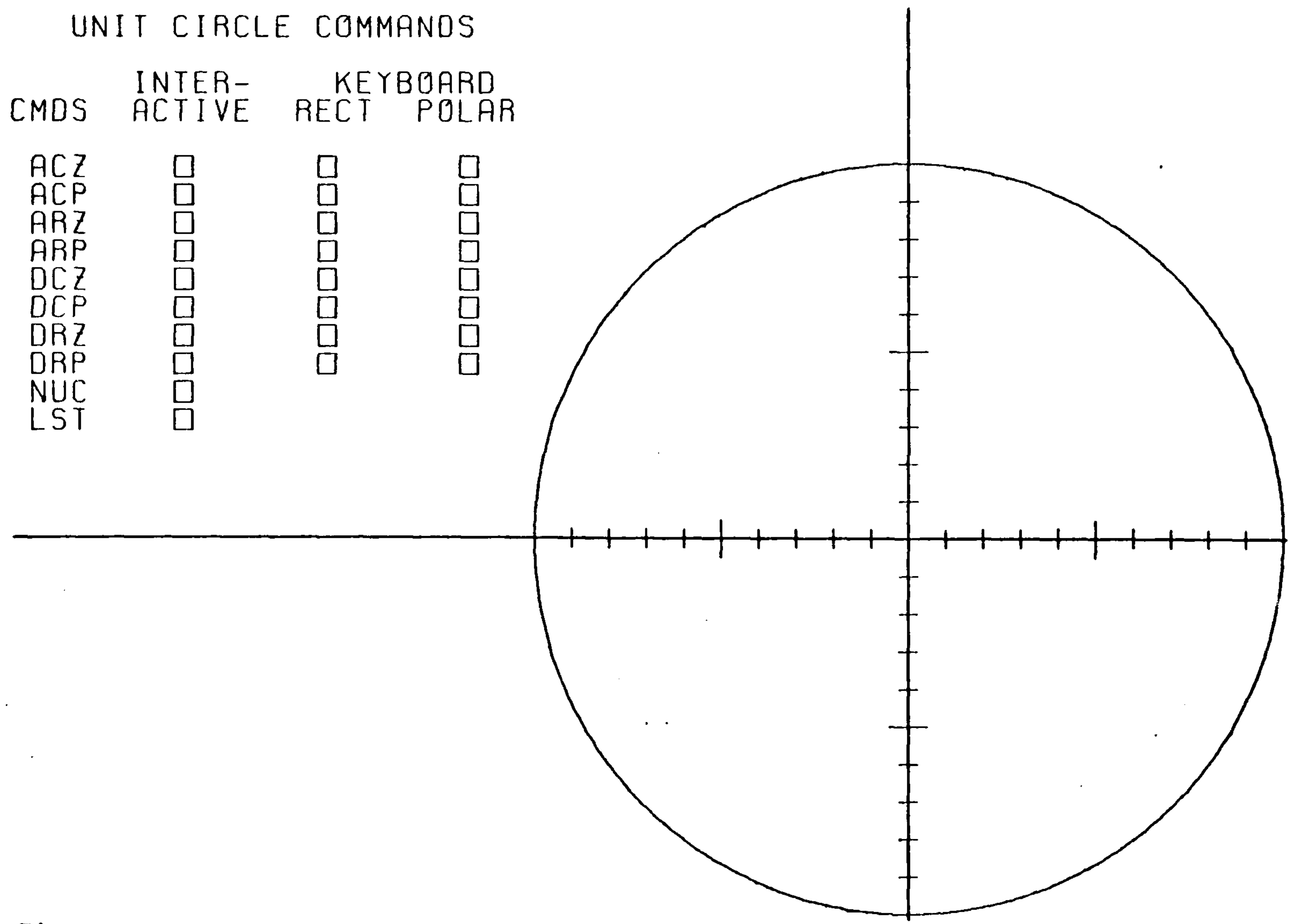


Figure 6-3 Trace Cursor Position Required To Place A Real Zero At The Origin

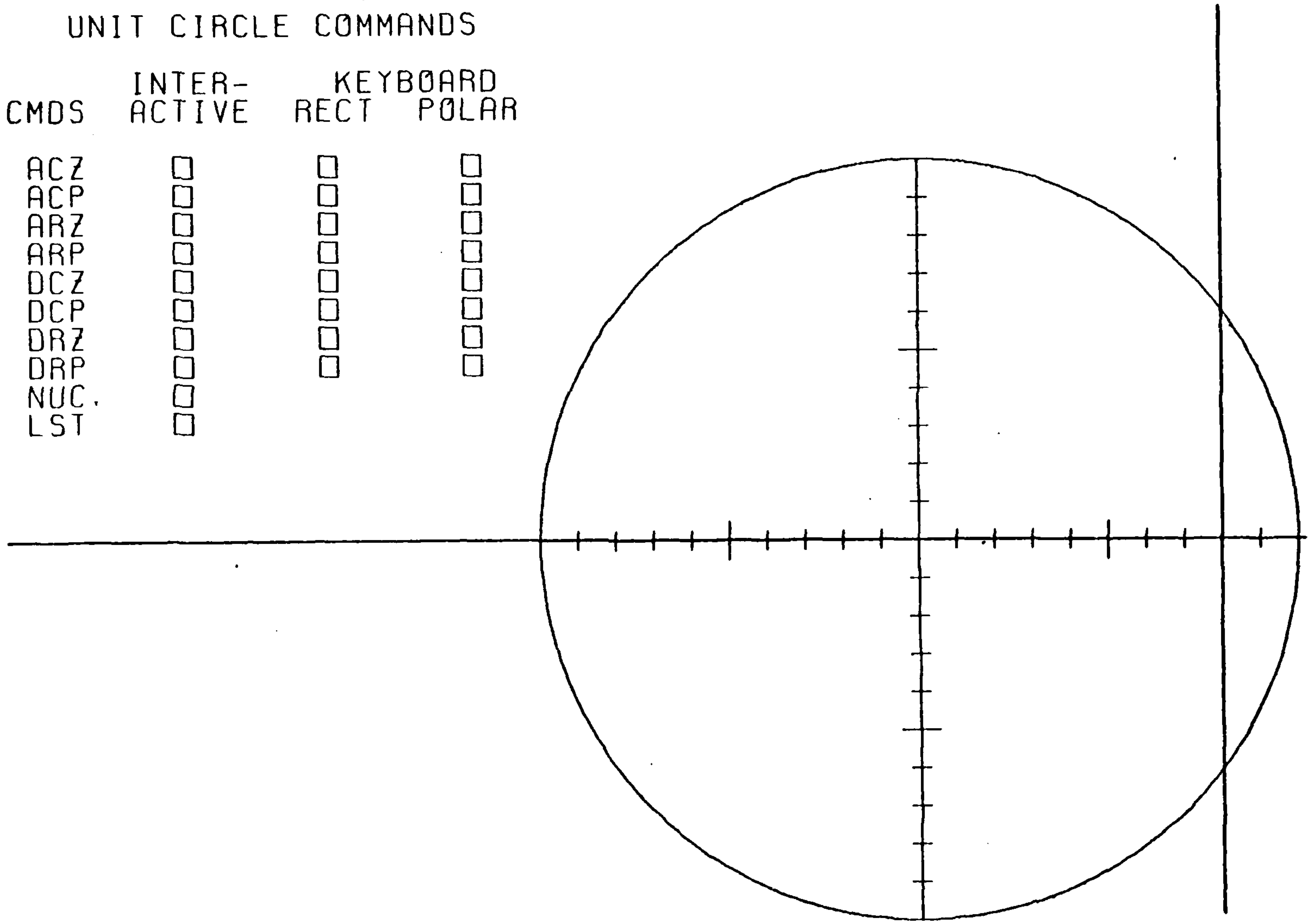


Figure 6-4 Trace Cursor Position Required To Put A Real Zero At (+0.8)

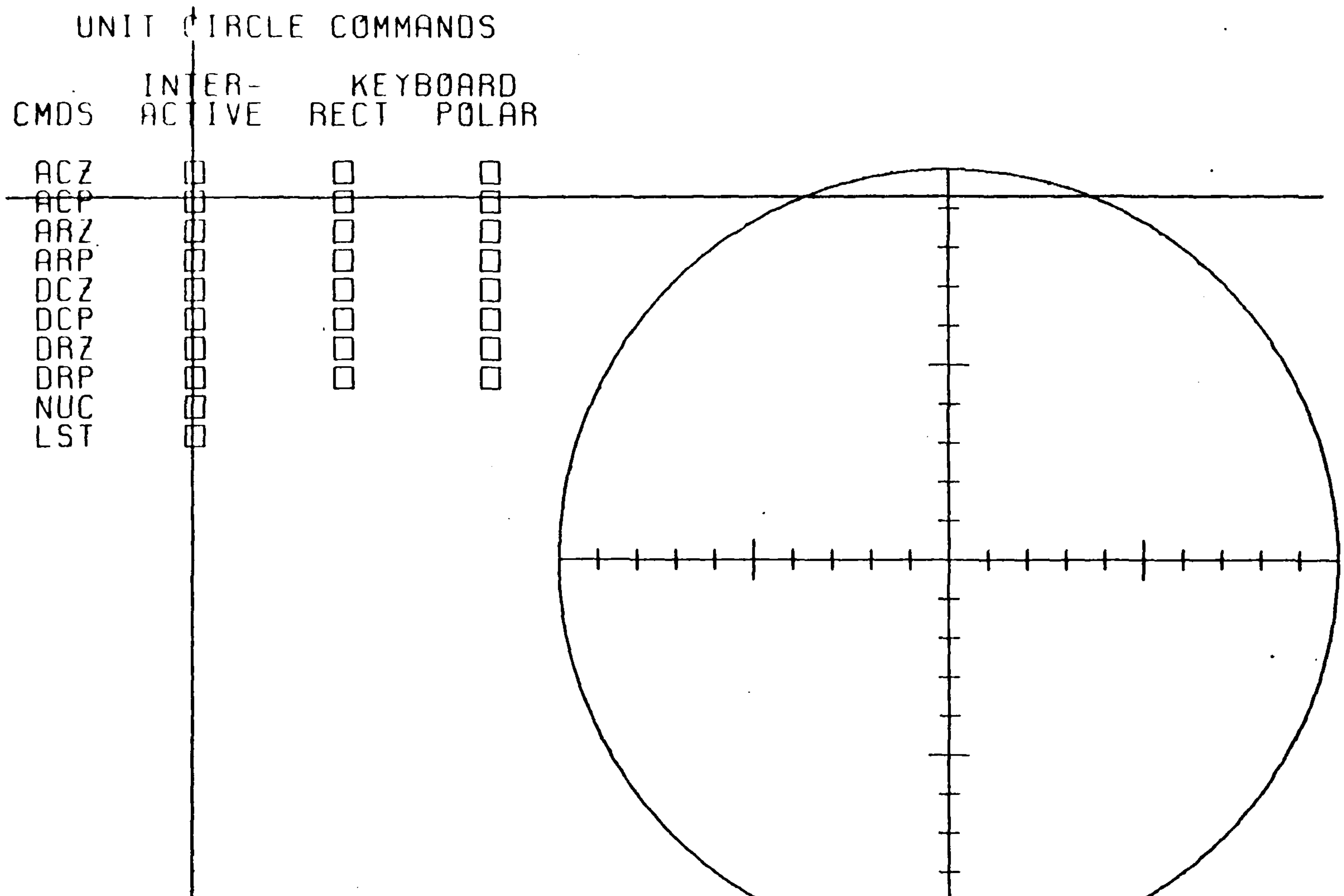


Figure 6-5 Trace Cursor Position Required To Select The Command "Add A Complex Pole"-Interactive Graphics Mode

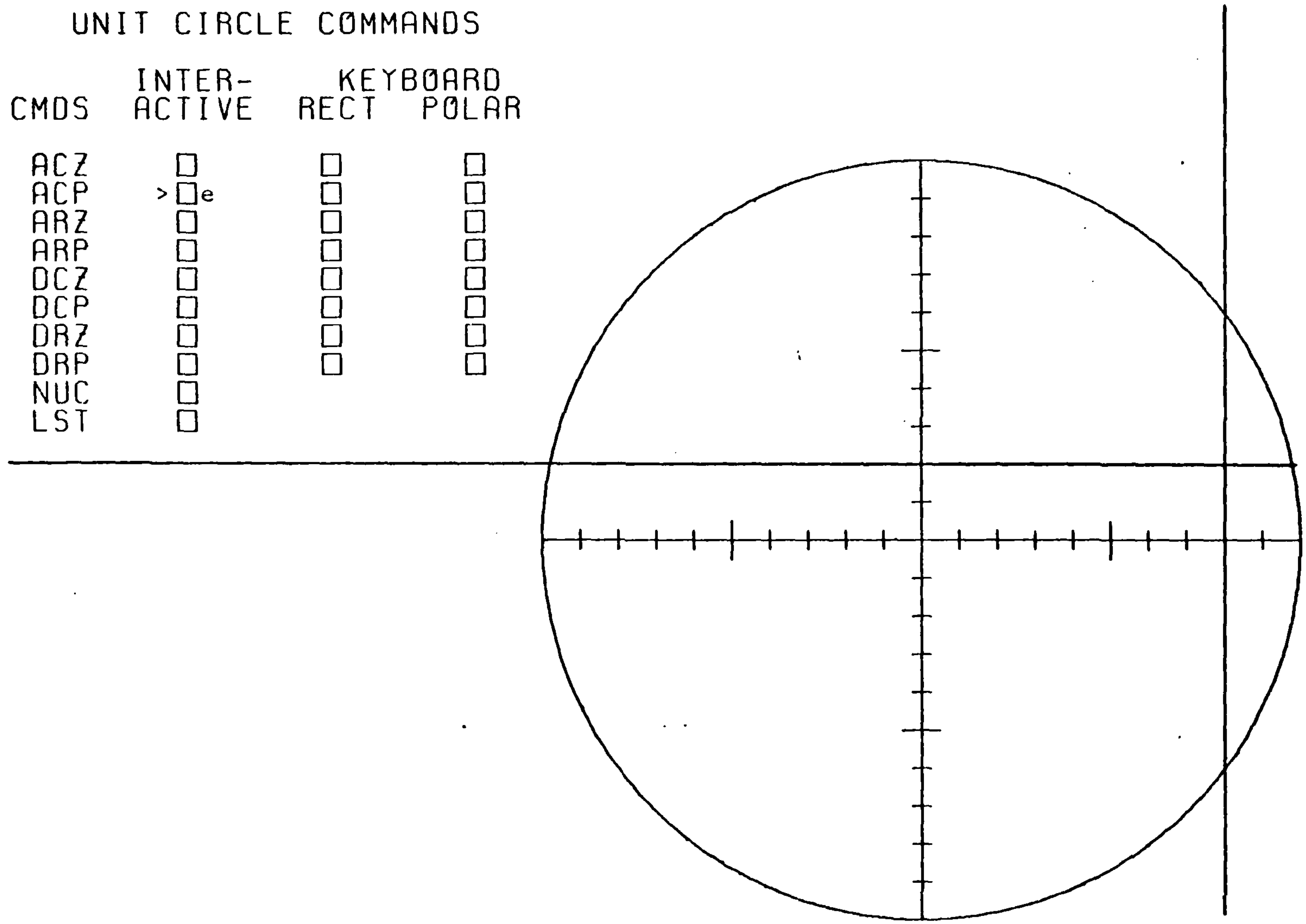


Figure 6-6 Trace Cursor Position Required To Enter A Complex Pole At (0.8+/-J0.2)

characteristics the command typed is:

RESPONSE

To get a hard copy of the filter responses the command:

EXEC HRDSCPY

is typed. Figures 6-7a, b, c, d, e, f, and g show the pertinent portions of the output.

3. EXAMPLE TWO - RECTANGULAR MODE

This second filter example is taken from [Ref. 5], page 222. The filter is a fourth-order low pass Chebyshev. The derivation of the transfer function is described in the text. The given transfer function is factored to give the desired pole/zero locations. The filter has two complex zeros at (-1.0 +/- j0.0) and two complex pole pairs, one pair at (.7498 +/- j.5348), and one pair at (.7774 +/- j.2120). The filter gain is given as 0.001836.

In a manner exactly paralleling the previous example, the user selects the command ACZRCT (which stands for Add a Complex Zero - Rectangular Coordinates), by positioning the trace cursor in the appropriate box, as in figure 6-8a. Once a keyboard character is typed, and the alphanumeric cursor moves into the box, the "space", "e", and "return" keys are typed. The ASDF program will erase the screen and draw the display shown in figure 6-8b. The user then types in the rectangular coordinates of either complex zero in the first complex zero pair, figure 6-8_. The system will

```
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z      REAL        Z   IMAGINARY   Z        RHO       Z      THETA     Z RT Z
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z     0.0          Z     0.0       Z   0.0            Z     0.0        Z RZ Z
Z     0.7999999    Z     0.0       Z   0.7999999      Z     0.0        Z RZ Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
Z    **********    Z   **********  Z  **********      Z   **********   Z    Z
ZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZP
P     0.7999999    P +/- 0.2000000 P   0.8246210      P +/- 0.2449785  P CP P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
P    **********    P   **********  P  **********      P   **********   P    P
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP

NUMBER OF REAL ZEROES:  2
NUMBER OF COMPLEX ZERO PAIRS:  0
NUMBER OF REAL POLES:  0
NUMBER OF COMPLEX POLE PAIRS:  1
FILTER GAIN:  1.00000000
```

Figure 6-7a Pole Zero Locations For Example Two

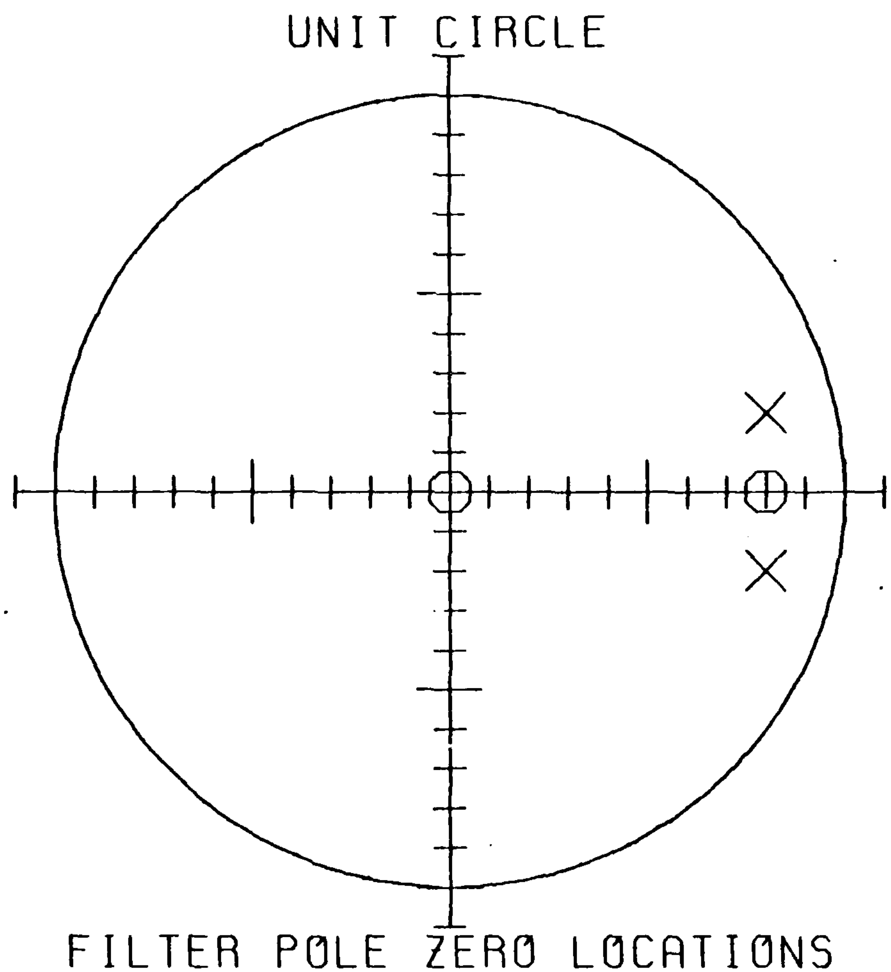


Figure 6-7b Pole Zero Locations For Example Two

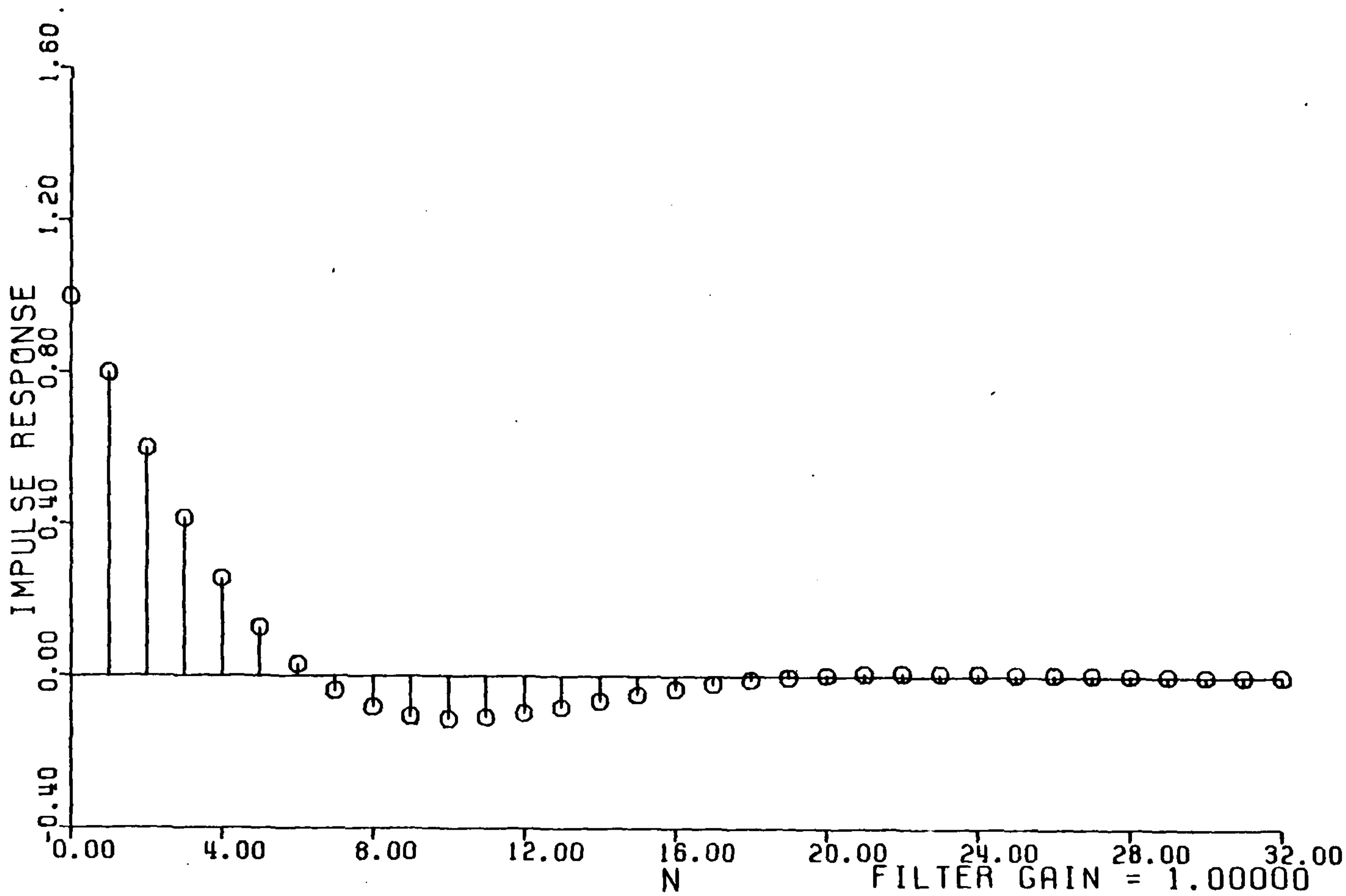


Figure 6-7c Unit Sample Response For Example Two

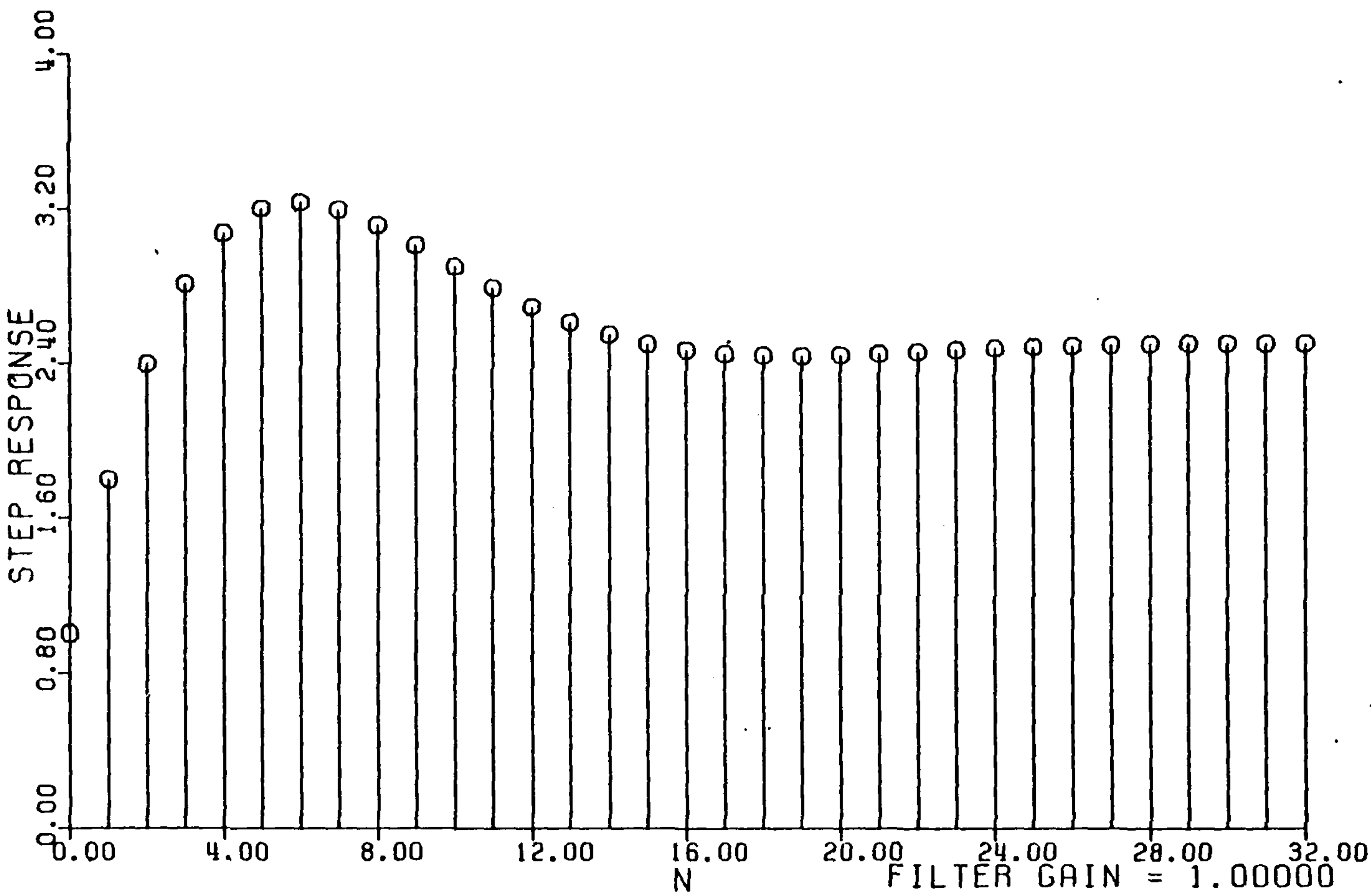


Figure 6-7d Step Response For Example Two

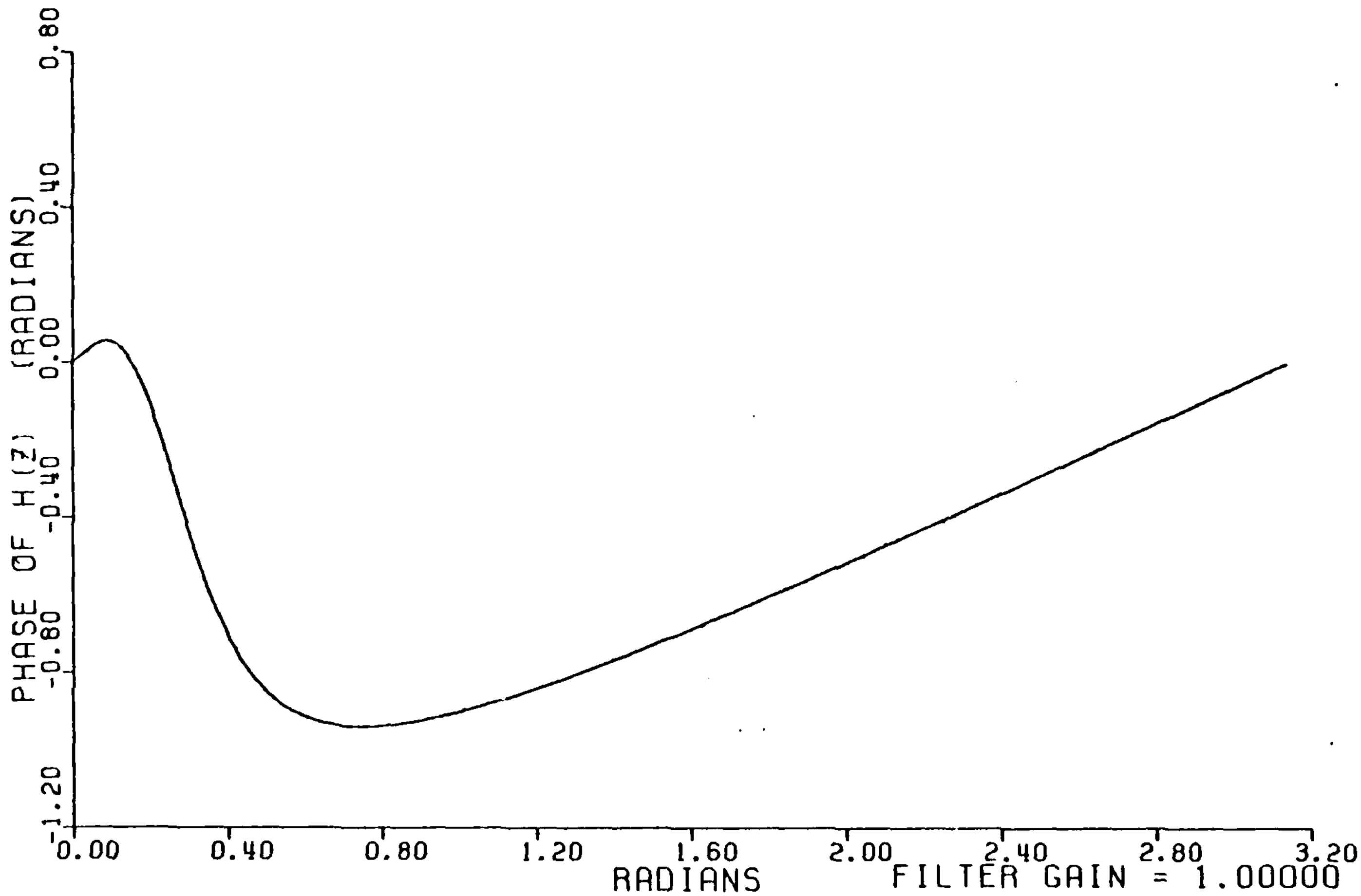


Figure 6-7e Phase Response For Example Two

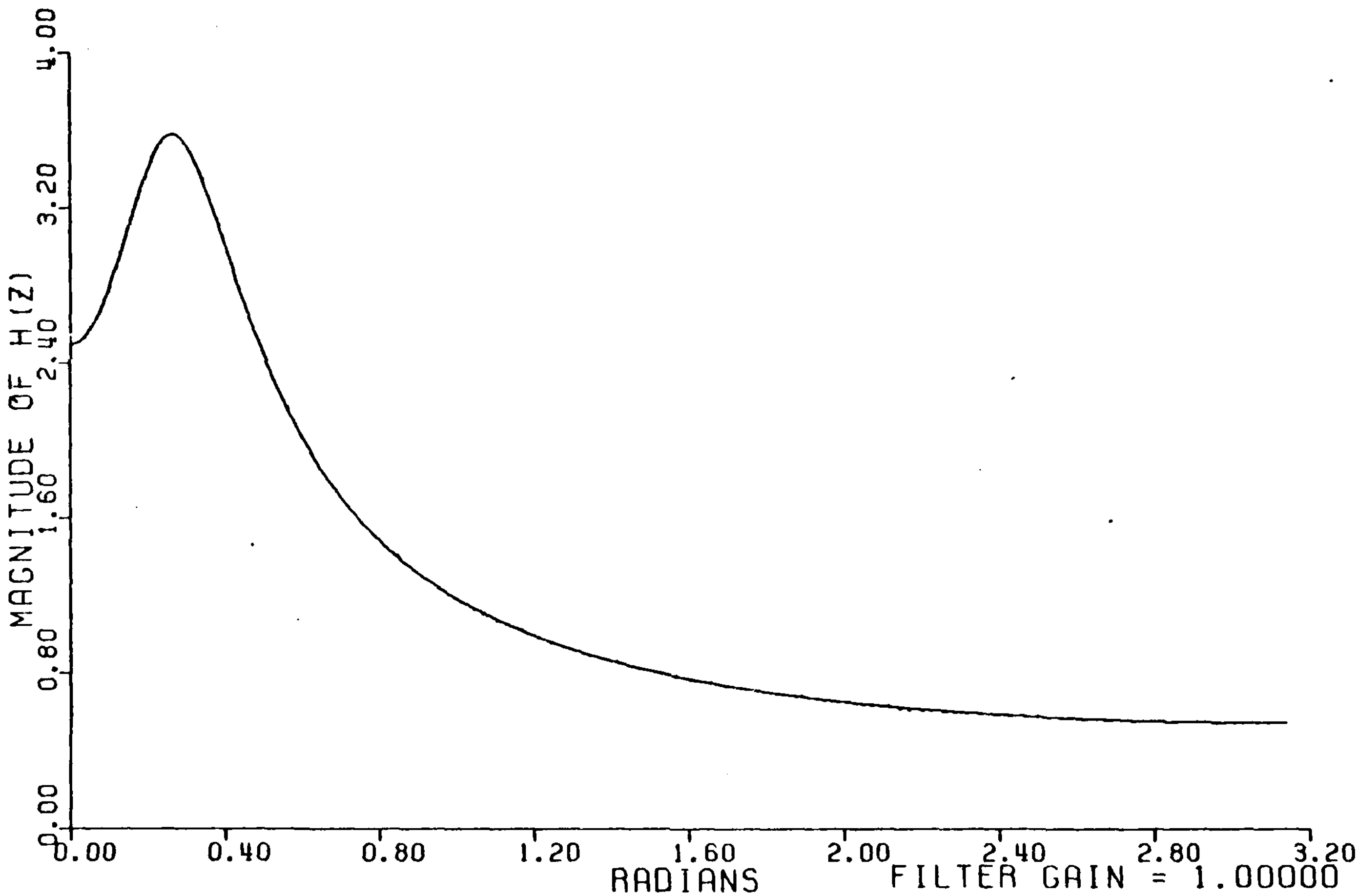


Figure 6-7f Magnitude Of H(z) For Example Two

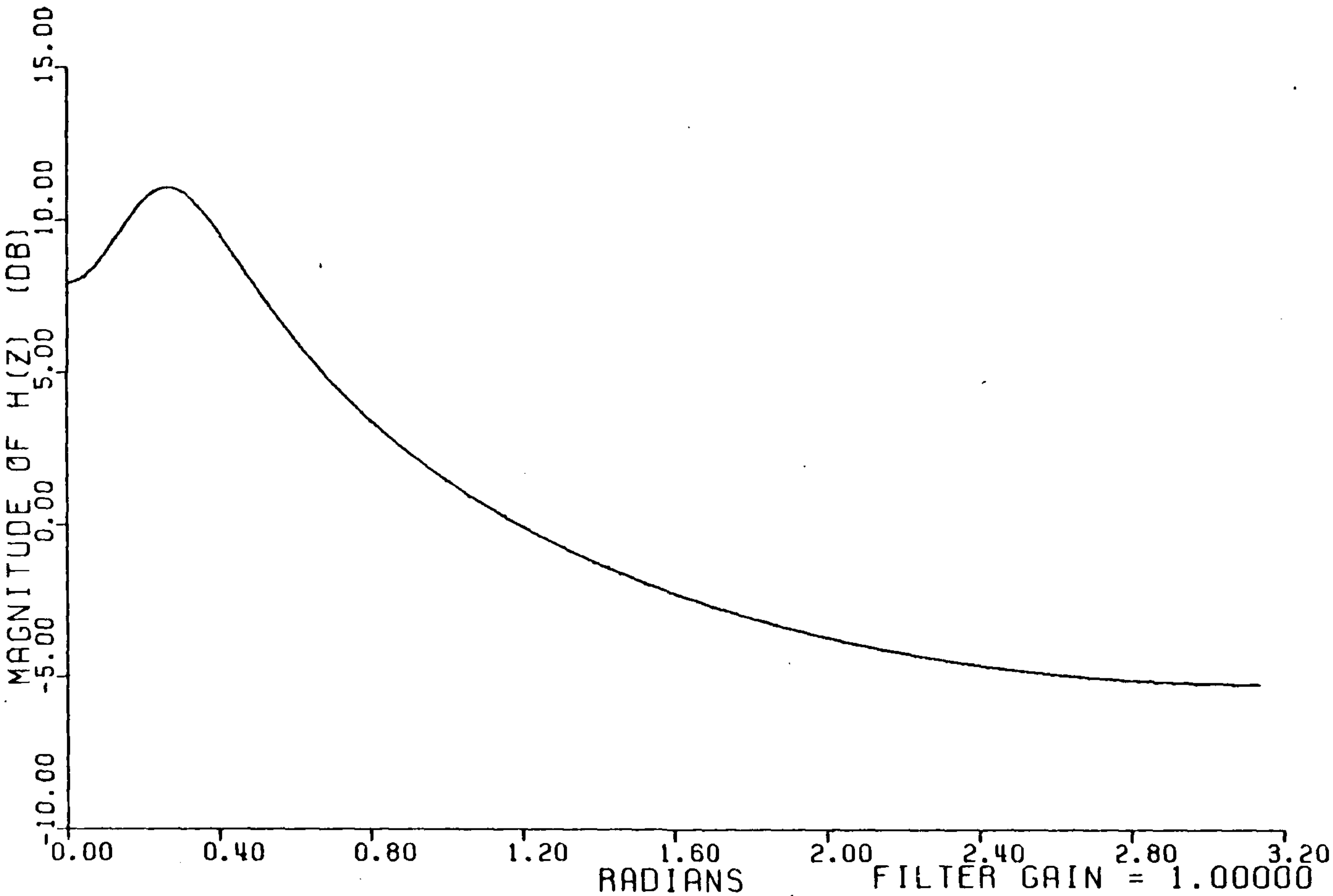


Figure 6-7g Magnitude Of The Transfer Function In Decibels For Example Two

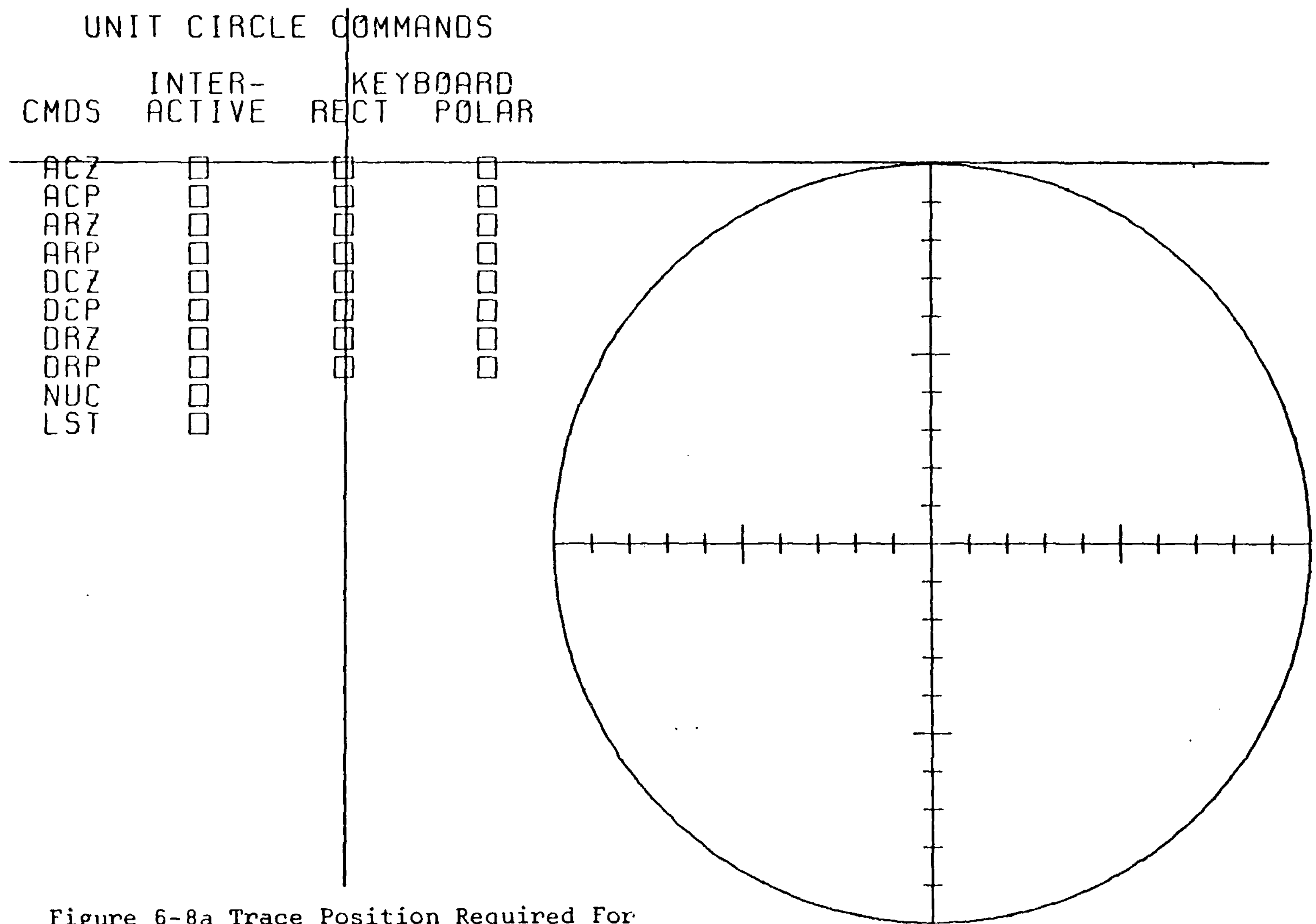


Figure 6-8a Trace Position Required For
Executing The Command "Add A Complex Zero"-Rectangular Coordinates Mode

```
TO ADD A COMPLEX ZERO TO THE
DISPLAY, ENTER THE REAL AND
IMAGINARY PARTS WITHIN THE BOXES
PROVIDED. EACH NUMBER REQUIRES A
DECIMAL. REAL PART MINUS SIGNS
MUST BE INCLUDED WITHIN THE BOX.
NOTE: ALL COMPLEX ZEROS MUST BE
WITHIN TEN UNITS OF THE ORIGIN.

     REAL            IMAGINARY
>>[          ]+/- J[          ]
```

Figure 6-8b Screen Display For The Command ACZRCT

```
TO ADD A COMPLEX ZERO TO THE
DISPLAY, ENTER THE REAL AND
IMAGINARY PARTS WITHIN THE BOXES
PROVIDED. EACH NUMBER REQUIRES A
DECIMAL. REAL PART MINUS SIGNS
MUST BE INCLUDED WITHIN THE BOX.
NOTE: ALL COMPLEX ZEROS MUST BE
WITHIN TEN UNITS OF THE ORIGIN.

     REAL            IMAGINARY
>>[  -1.0   ]+/- J[   0.0    ]

THE NEW COMPLEX ZERO WILL BE -1.0000000 +/- J 0.0
     IF CORRECT TYPE 'Y', IF NOT 'N'
```

Figure 6-8c Screen Display After Entering The First Complex Zero

ask whether or not the root location has been correctly entered. The user responds 'y' for yes, 'n' for no. If the root has been incorrectly entered, an 'n' response will rewrite the screen for another attempt. Upon receiving a 'y' response ASDF rewrites the unit circle and command array, and plots the new root. The second complex zero is entered exactly as the first.

The complex poles are entered as were the zeros. The trace cursor is first placed in the box ACPRCT. (Add a Complex Pole - Rectangular Coordinates), as in figure 6-9. Figure 6-10 shows the unit circle after all poles and zeros have been entered. The POLZRO command is terminated by aligning the trace cursor with any box, typing a character, waiting for the flashing alphanumeric cursor to enter the box, and typing "space", "x". The system responds by asking about the filter gain. In this example the filter gain is to be changed to .001836, so the correct response is 'y'. The new filter gain, 0.001836, is typed to terminate the POLZRO command.

To view the filter responses, the user enters the command:

RESPONSE

when the *ASDF-RE signature appears. The hard copy responses of the filter are shown in figures 6-11a through 6-11g.

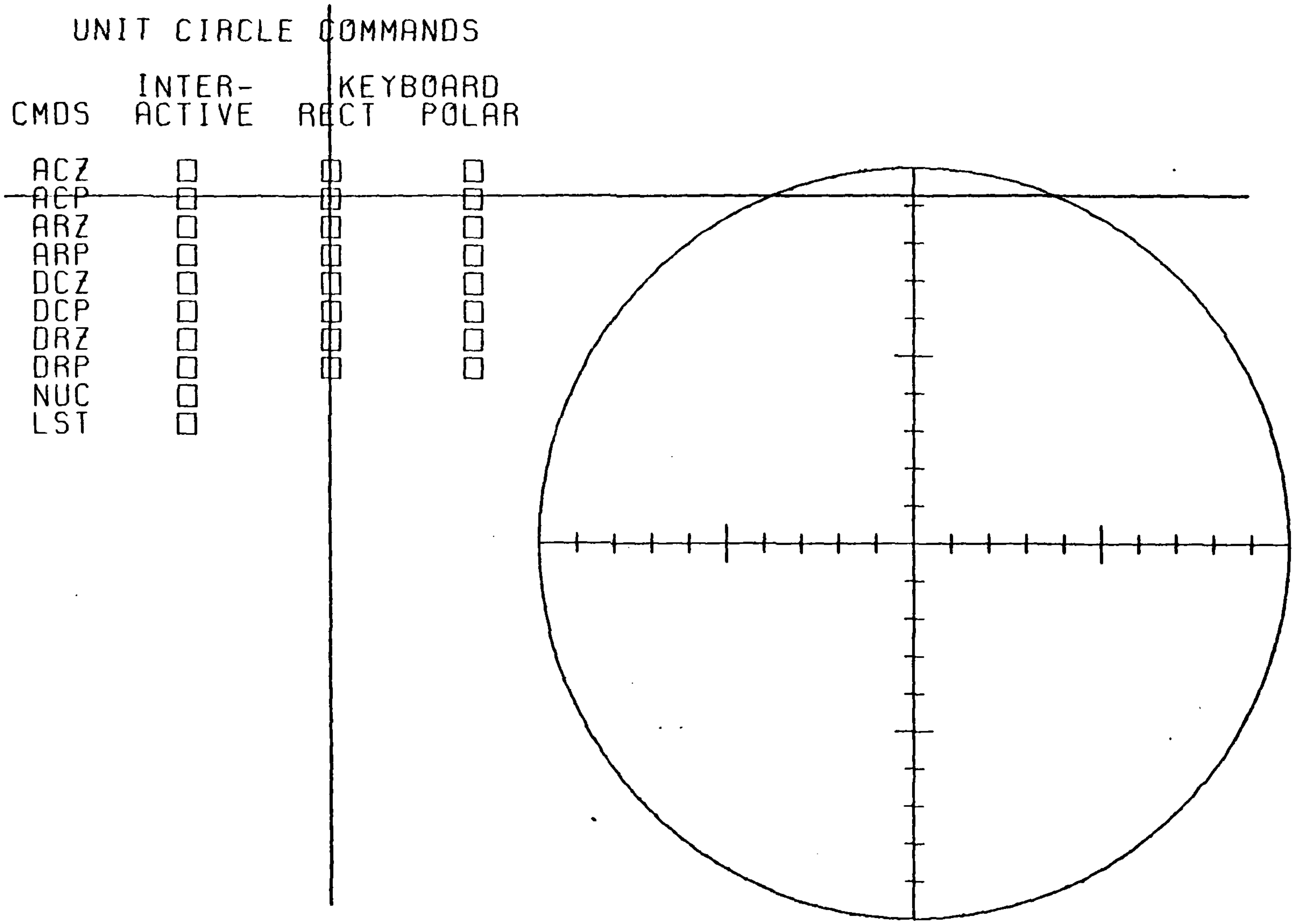


Figure 6-9 Trace Cursor Position To Execute Command ACPRCT

UNIT CIRCLE COMMANDS

CMDS	INTER-ACTIVE	KEYBOARD RECT	KEYBOARD POLAR
ACZ	☐	☐	☐
ACP	☐	☐	☐
ARZ	☐	☐	☐
ARP	☐	☐	☐
DCZ	☐	☐	☐
DCP	☐	☐	☐
DRZ	☐	☐	☐
DRP	☐	☐	☐
NUC	☐		
LST	☐		

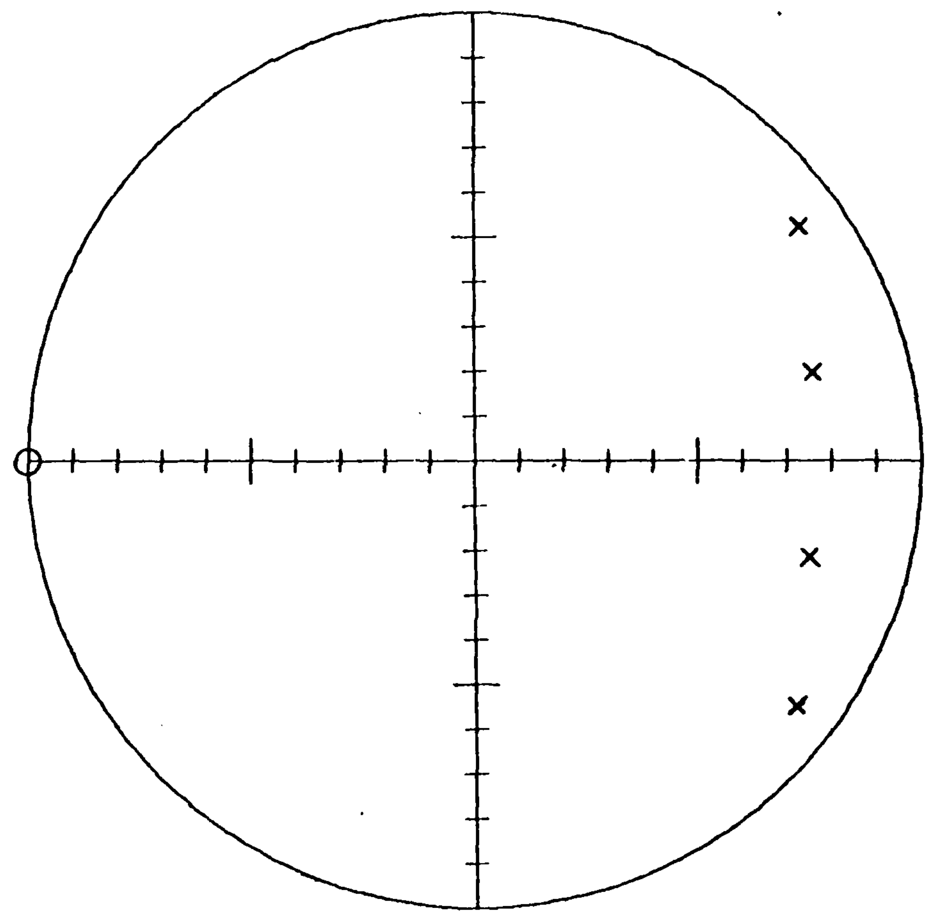


Figure 6-10 Unit Circle After All Poles and Zeros Have Been Returned-Example Two

```
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z      REAL        Z   IMAGINARY   Z        RHO       Z      THETA      Z RT Z
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z    -1.0000000    Z +/- 0.0       Z    1.0000000     Z +/- 3.1415930 Z CZ Z
Z    -1.0000000    Z +/- 0.0       Z    1.0000000     Z +/- 3.1415930 Z CZ Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
Z    **********    Z     **********Z  **********      Z    ********** Z    Z
ZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZP
P     0.7498000    P +/- 0.5347999 P    0.9209836     P +/- 0.6195678 P CP P
P     0.7774000    P +/- 0.2119999 P    0.8057882     P +/- 0.2662303 P CP P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
P    **********    P     **********P  **********      P    ********** P    P
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP

NUMBER OF REAL ZEROES:  0
NUMBER OF COMPLEX ZERO PAIRS:  2
NUMBER OF REAL POLES:  0
NUMBER OF COMPLEX POLE PAIRS:  2
FILTER GAIN:  0.00180000
```

Figure 6-11a Pole Zero Locations For Example Two

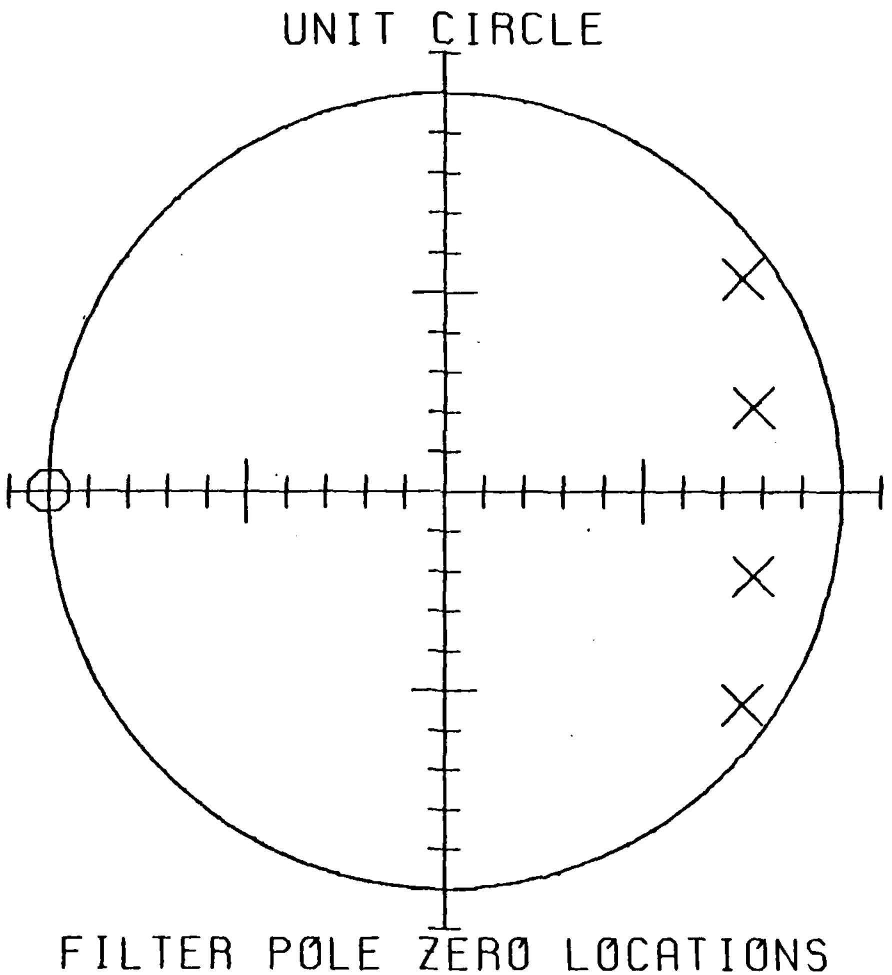


Figure 6-11b Pole Zero Locations For Example Two

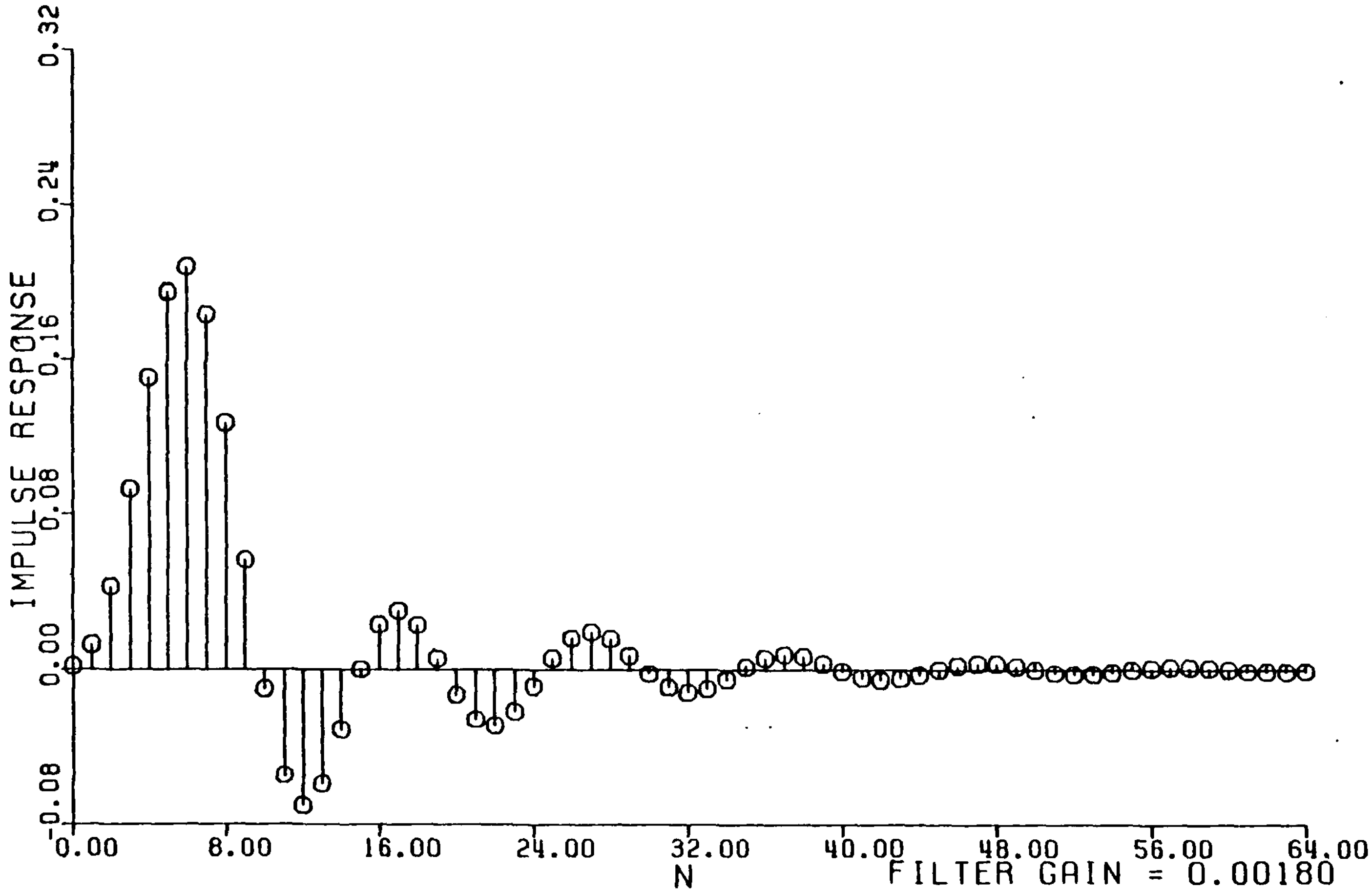


Figure 6-11c Unit Sample Response For Example Two

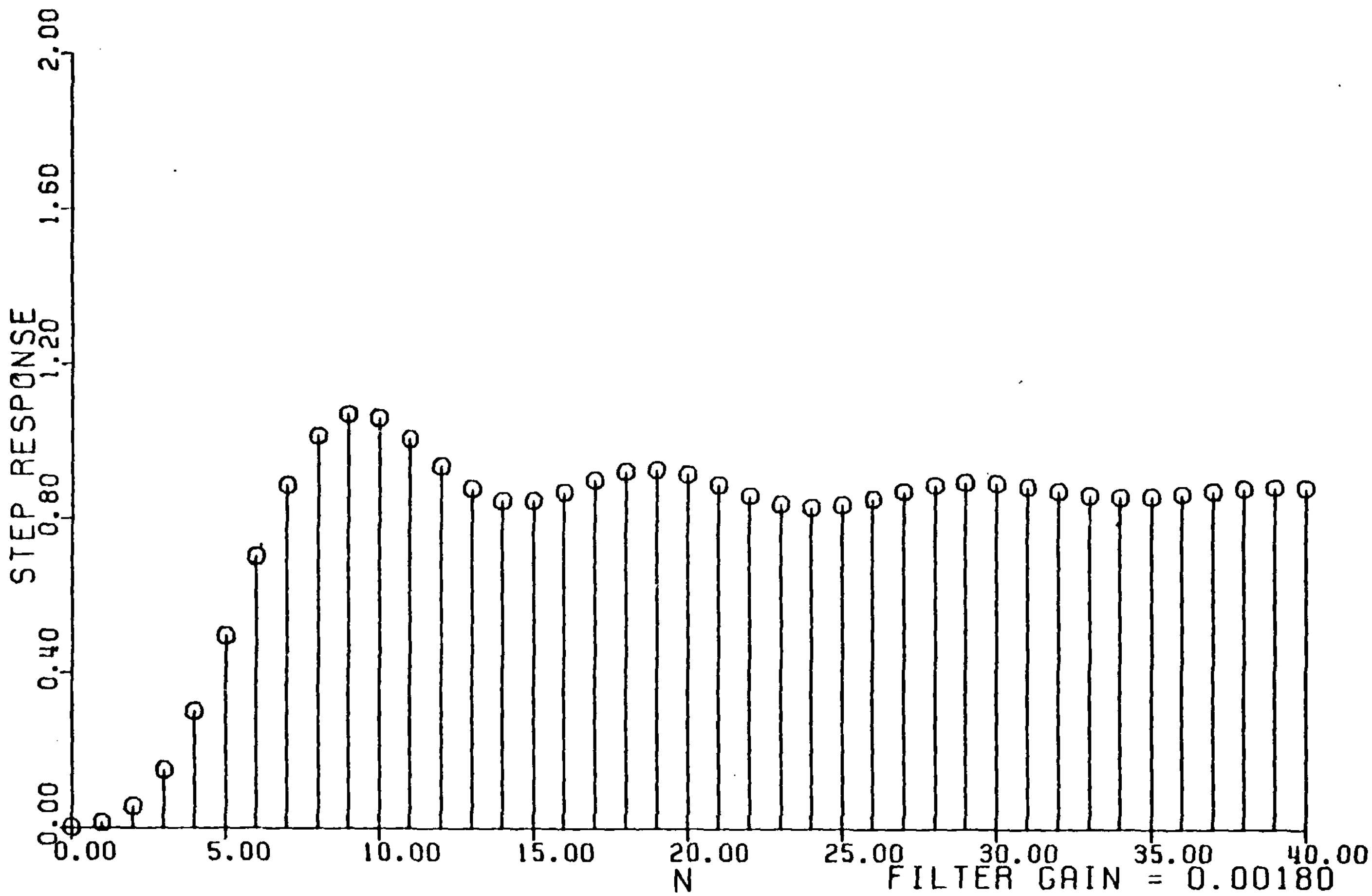


Figure 6-11d Unit Step Response For Example Two

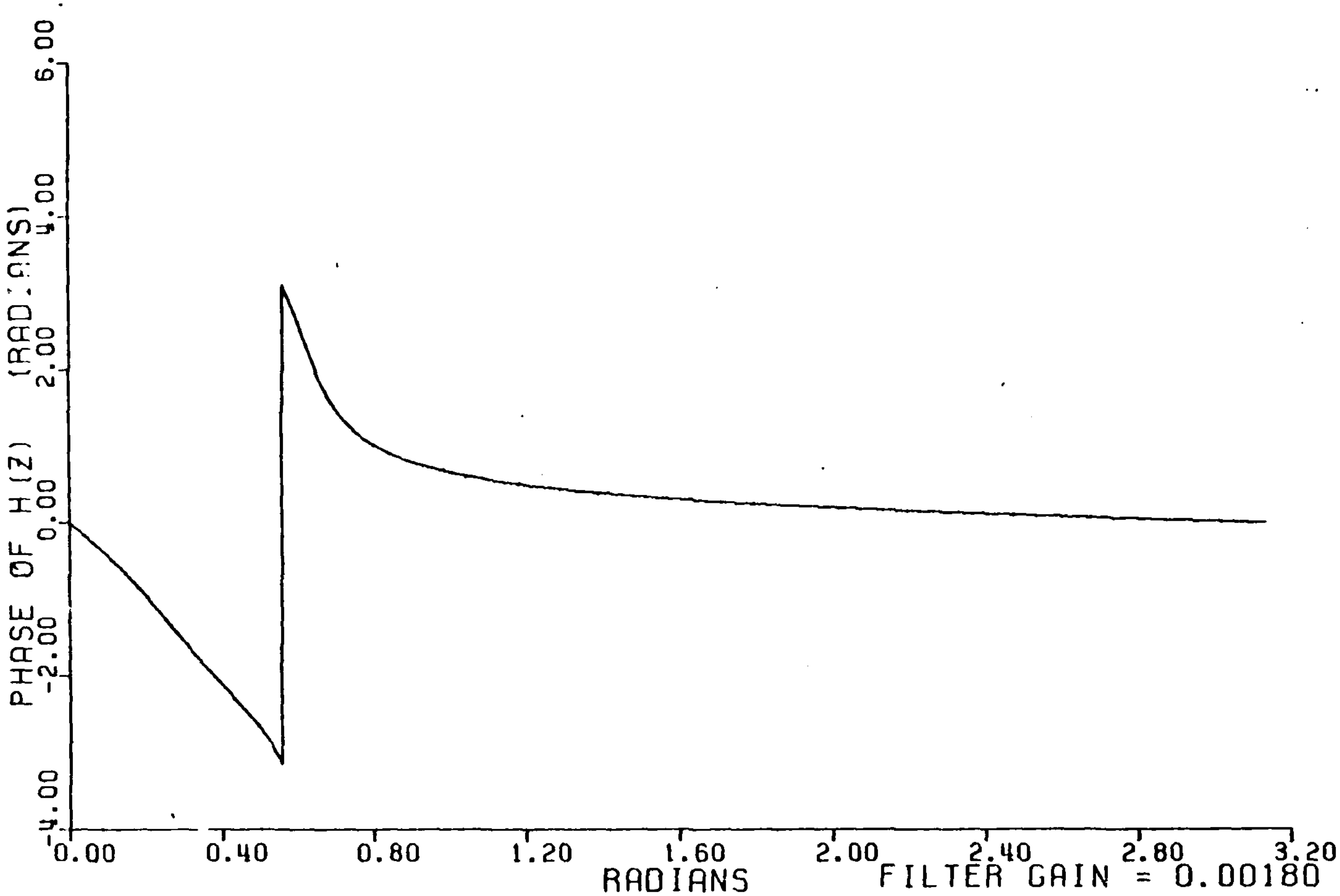


Figure 6-11e Phase Response For Example Two

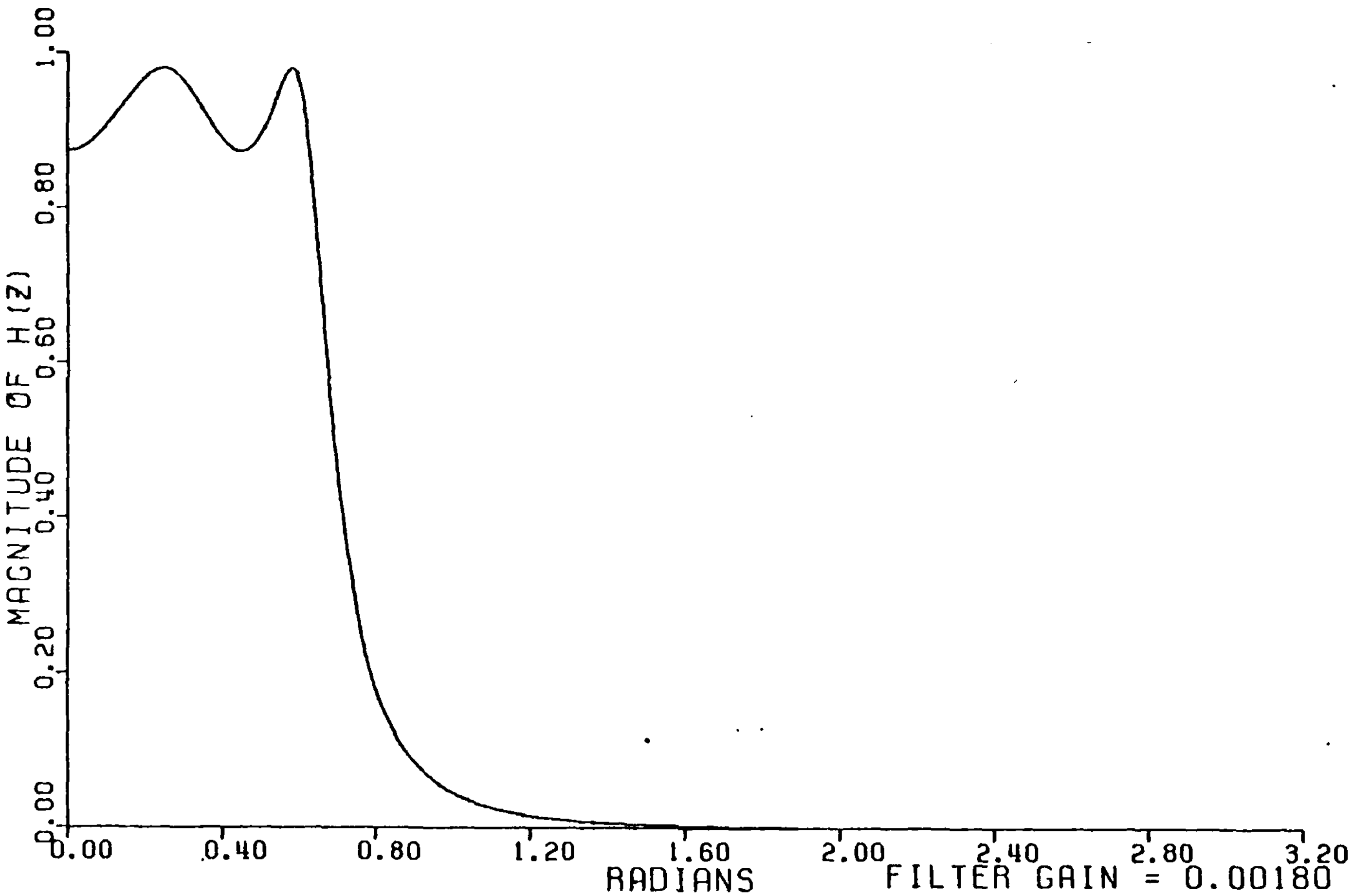


Figure 6-11f Magnitude of H(z) For Example Two

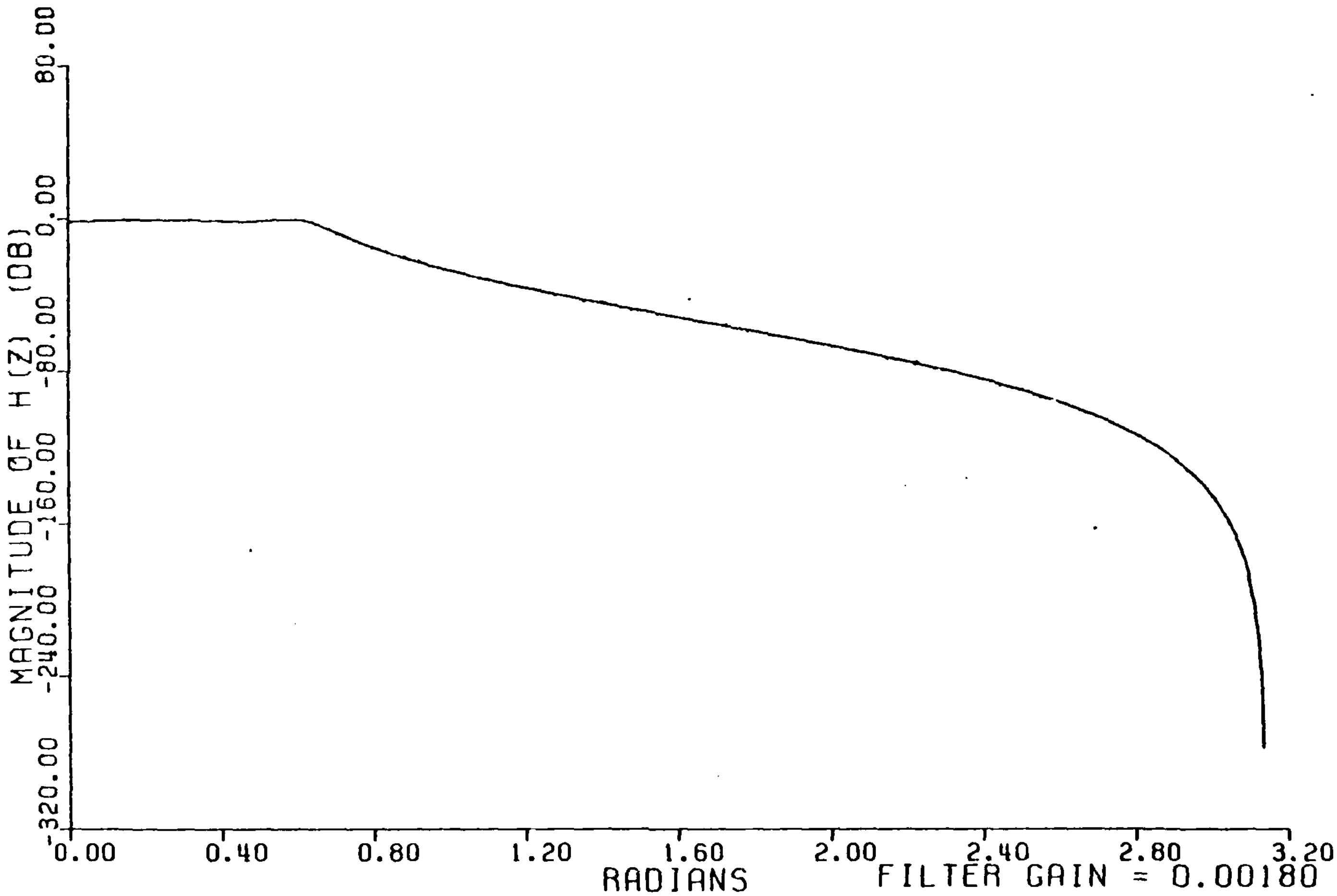


Figure 6-11g Magnitude Of H(z) In Decibels For Example Two

C. EXAMPLE THREE - POLAR MODE

This example uses the polar mode to generate an all pass filter. The filter complex pole pairs are to be located at (0.2 at +/- 45 degrees), and (0.5 at +/- 30 degrees). The filter zeros are to be at (5.0 at +/- 45 degrees), and (2.0 at +/- 30 degrees). The filter gain is to be 0.01. The first step is to change the specifications in degrees to radians. Thirty degrees equals .5235 radians. Forty five degrees is .7854 radians. The entry of poles and zeros exactly parallels the above examples, however, the desired comands are ACPPLR, (Add a Complex Pole - Polar Coordinates), and ACZPLR, (Add a Complex Zero - Polar Coordinates). Figure 6-12a shows the trace cursor position for ACPPLR. Figure 6-12b shows the trace cursor position for ACZPLR. The angular part of any pole or zero entered in the polar mode must be in radians. Figure 6-12c shows the entry of the first complex zero pair. Figure 6-12d shows the entry of the first complex pole pair. Figure 6-12e shows the change in filter gain from 1.00 to 0.01. Figures 6-13a through 6-13g show the hard copy responses for this filter. It should be noted that zeros which plot outside of the unit circle are indicated by stars. All such zeros are plotted at a radius of 1.1 along the same angular component as the actual zero location. Zeros are plotted this way for two reasons: first to make the output display consistent for all relevant filters, and second to emphasize zeros which

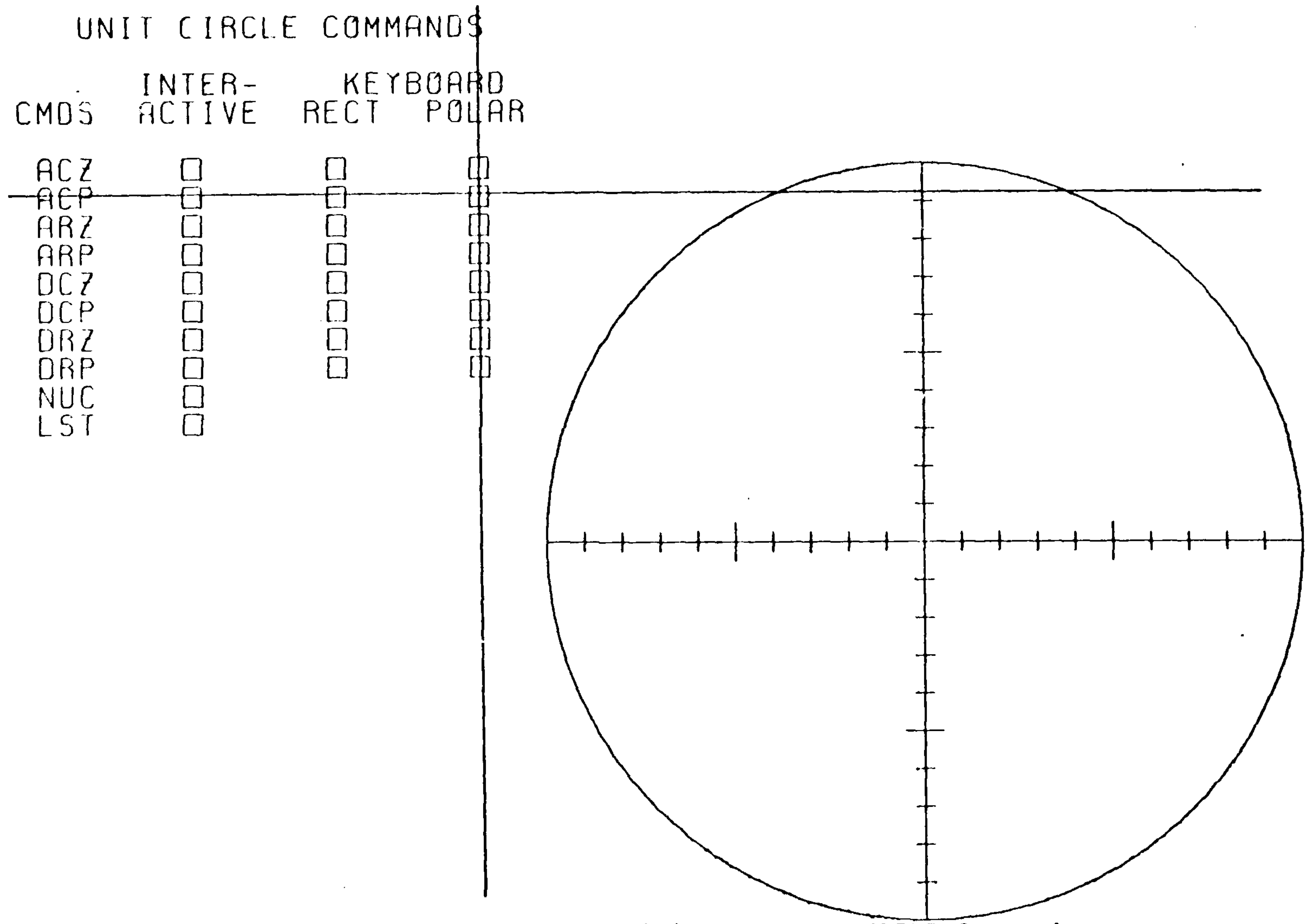


Figure 6-12a Trace Cursor Position For The ACPPLR Command

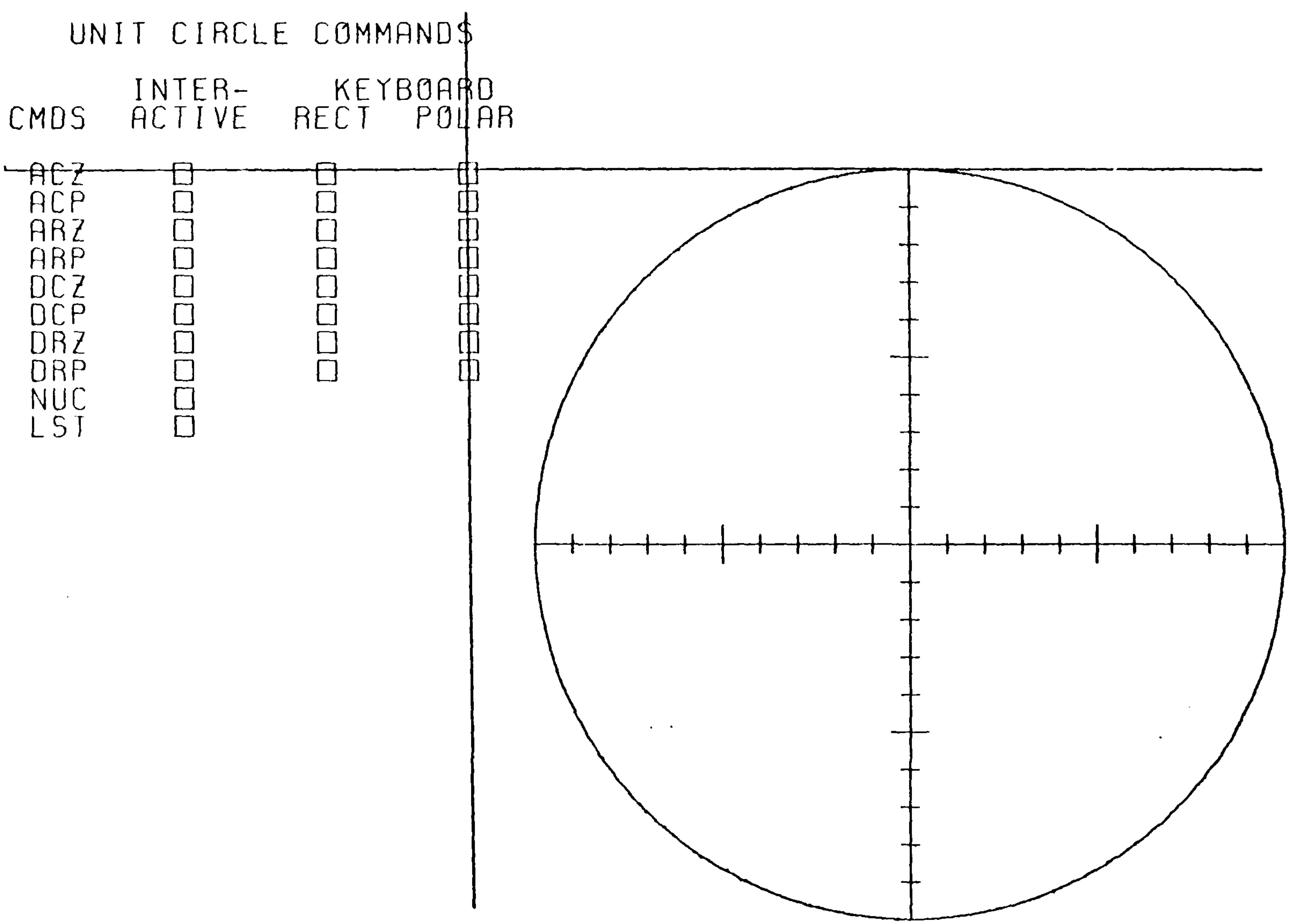


Figure 6-12b Trace Cursor Position For The ACZPLR Command

```
TO ADD A COMPLEX ZERO TO THE
DISPLAY, ENTER THE RADIAL AND
THETA COMPONENTS WITHIN PROVIDED
BOXES.  THE MAGNITUDE OF RHO MUST
BE LESS THAN TEN, THETA(RADIANS)
EACH NUMBER REQUIRES A DECIMAL,
AND AS APPROPRIATE A MINUS SIGN.
BOTH ARE CONFINED TO THE BOXES.

           MAGNITUDE          ANGLE
        >>[   15.0   ]+/-∡[.7854     ]

THE NEW COMPLEX ZEROS WILL BE 15.0000000 +/-∡0.7854000
        IF CORRECT TYPE 'Y', IF NOT 'N'.
```

Figure 6-12c Screen Display After Entering The First Complex Zero Pair In Polar Coordinates

```
TO ADD A COMPLEX POLE WITHIN THE
UNIT CIRCLE ENTER THE RADIAL AND
THETA COMPONENTS WITHIN PROVIDED
BOXES.  RHO MUST BE ONE OR LESS.
THETA MUST BE ENTERED IN RADIANS
EACH NUMBER REQUIRES A DECIMAL,
AND AS APPROPRIATE A MINUS SIGN.
BOTH ARE CONFINED TO THE BOXES.

          MAGNITUDE        ANGLE
        >>[   0.2   ]+/-∠[ .7854    ]

THE NEW COMPLEX POLES WILL BE 0.20000000 +/-∠0.7854000
          IF CORRECT TYPE 'Y', IF NOT 'N'.
```

Figure 6-12d Screen Display After Entering The First Complex Pole Pair In Polar Coordinates

```
THE FILTER GAIN IS CURRENTLY     1.00000000

DO YOU WISH TO CHANGE THE FILTER GAIN?

(TYPE Y OR N)

TYPE IN THE NEW FILTER GAIN 12 OR FEWER NUMBERS
INCLUDE A DECIMAL POINT

>0.01

NEW FILTER GAIN IS:  0.01000000
```

Figure 6-12e Screen Representation After Changing Filter Gain

```
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z      REAL        Z   IMAGINARY    Z       RHO        Z      THETA      Z RT Z
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z     3.5355272    Z +/- 3.5355396 Z    5.0000000      Z +/- 0.7854000 Z CZ  Z
Z     1.7321472    Z +/- 0.9998287 Z    2.0000000      Z +/- 0.5235003 Z CZ  Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
Z    **********    Z     ********** Z  **********      Z     ********** Z     Z
ZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZP
P     0.1414211    P +/- 0.1414216 P    0.2000000      P +/- 0.7854000 P CP  P
P     0.4330373    P +/- 0.2499570 P    0.5000000      P +/- 0.5235000 P CP  P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
P    **********    P     ********** P  **********      P     ********** P     P
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP

NUMBER OF REAL ZEROES:  0
NUMBER OF COMPLEX ZERO PAIRS:  2
NUMBER OF REAL POLES:  0
NUMBER OF COMPLEX POLE PAIRS:  2
FILTER GAIN:  0.01000000
```

Figure 6-13a Pole Zero Locations For Example Three

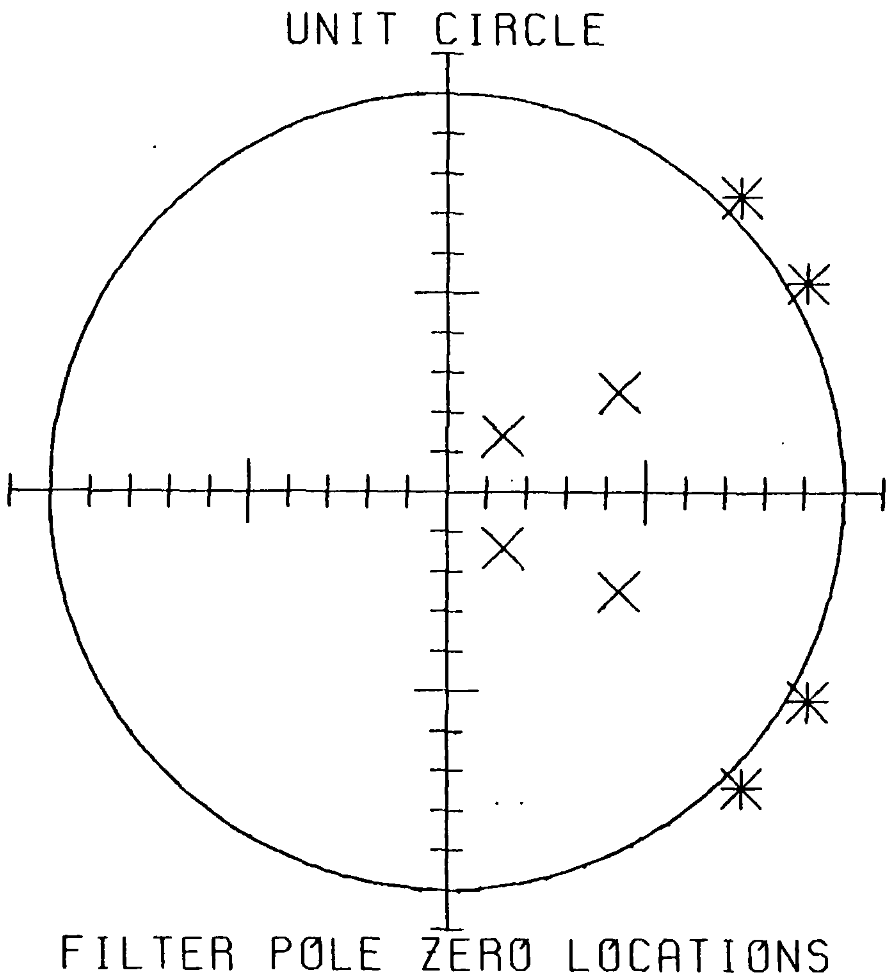


Figure 6-13b Pole Zero Location For Example Three

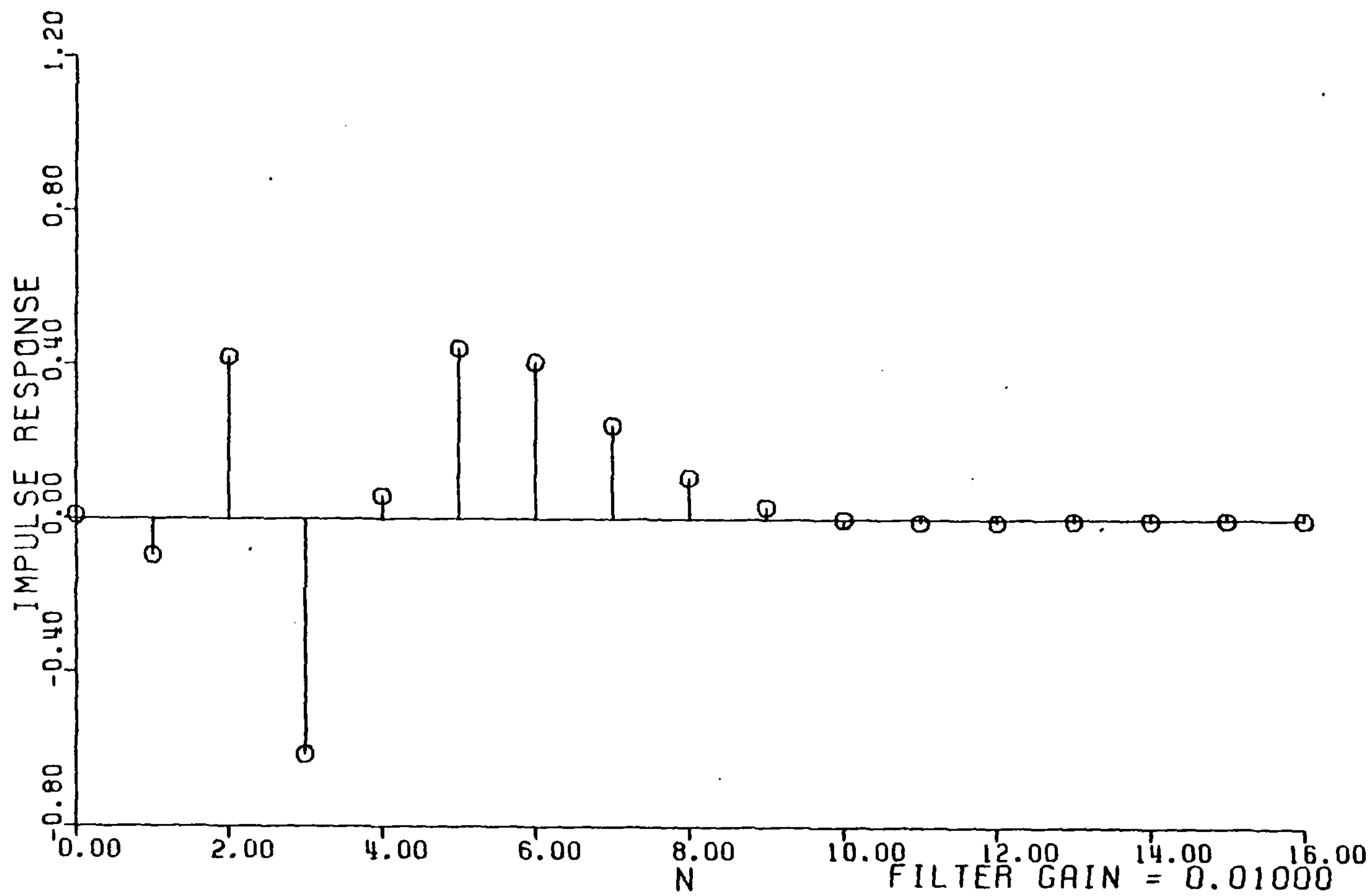


Figure 6-13c Unit Sample Response For Example Three

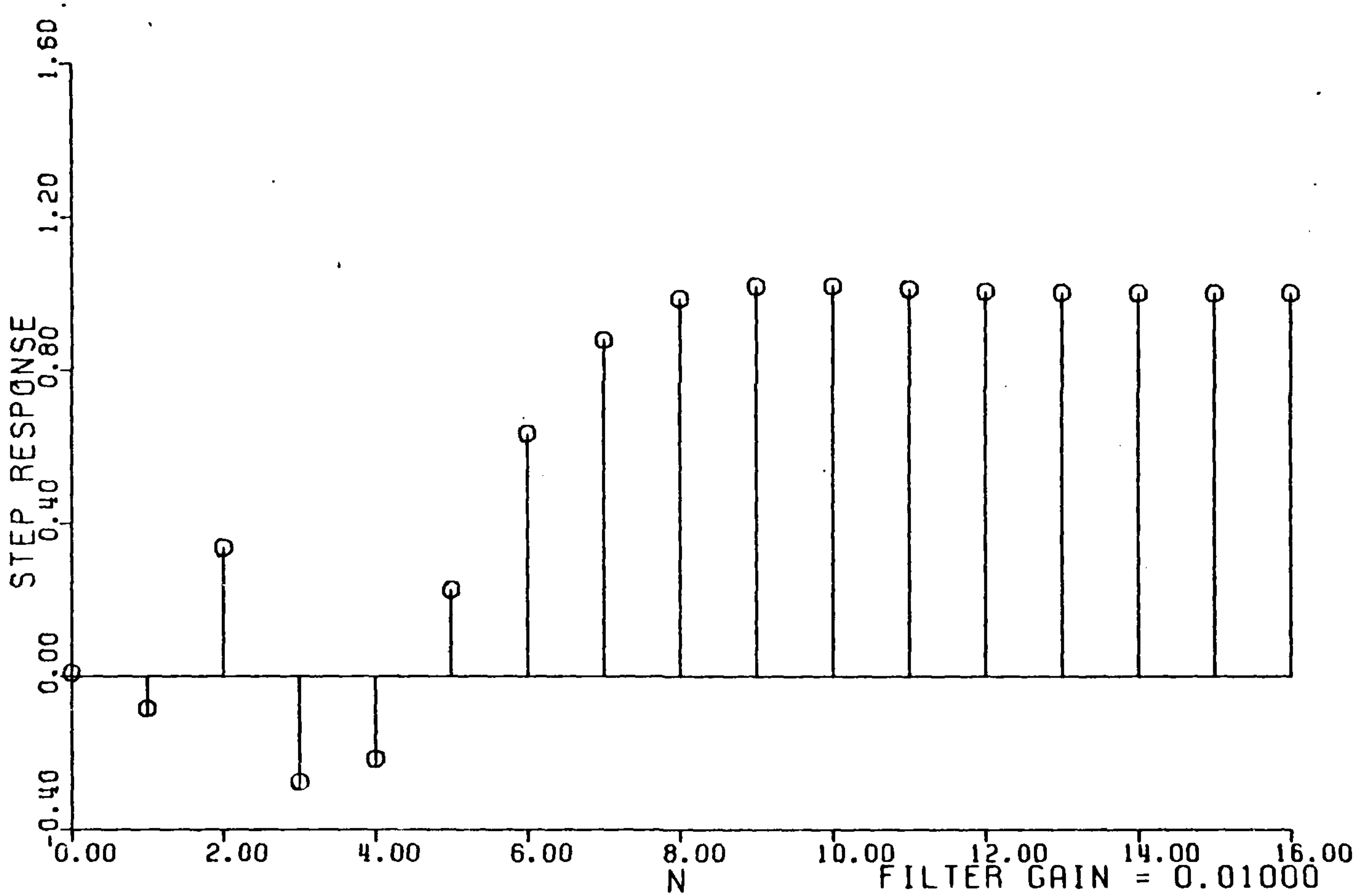


Figure 6-13d Unit Step Response For Example Three

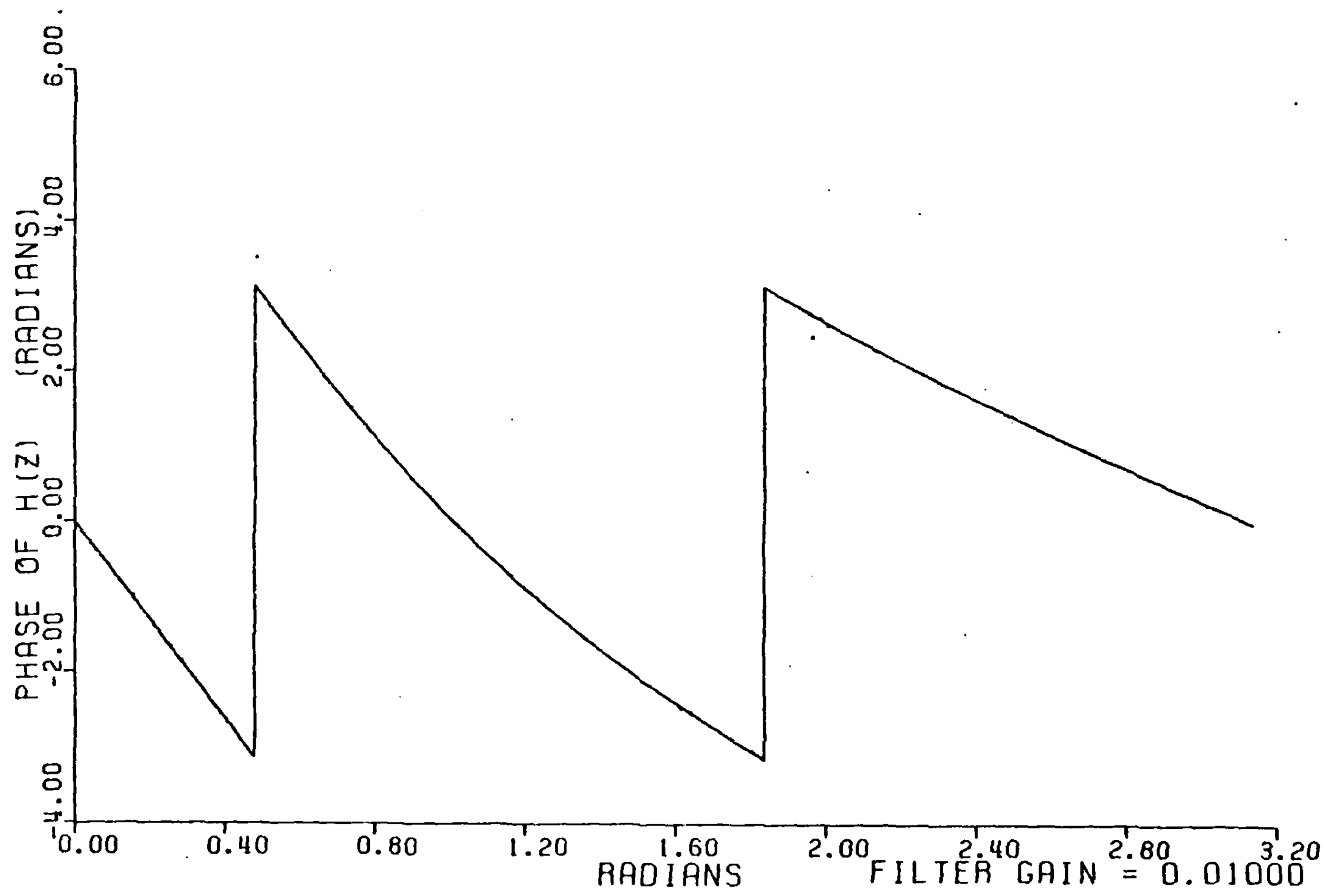


Figure 6-13e Phase Response For Example Three

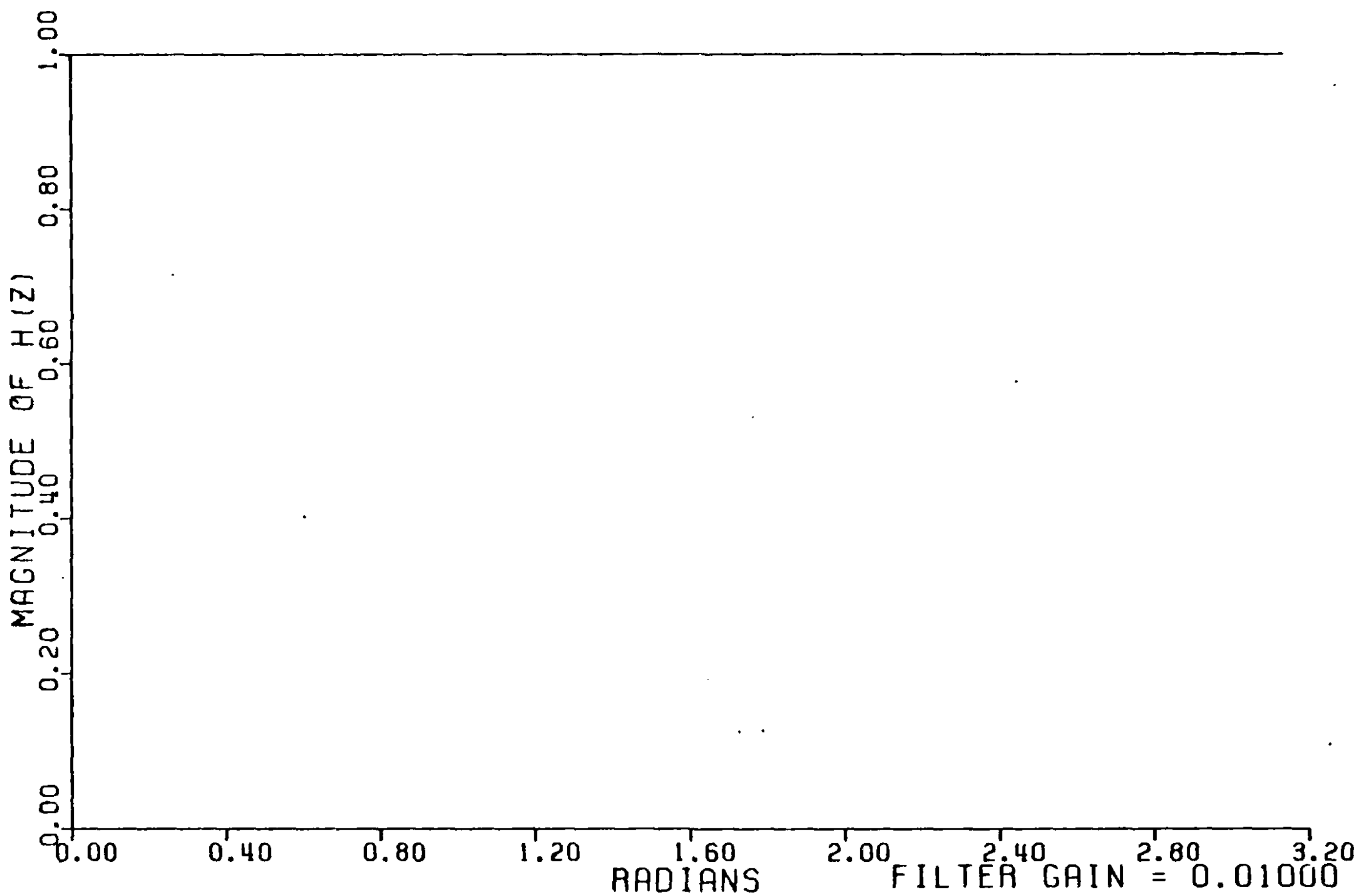


Figure 6-13f Magnitude Of H(z) For Example Three

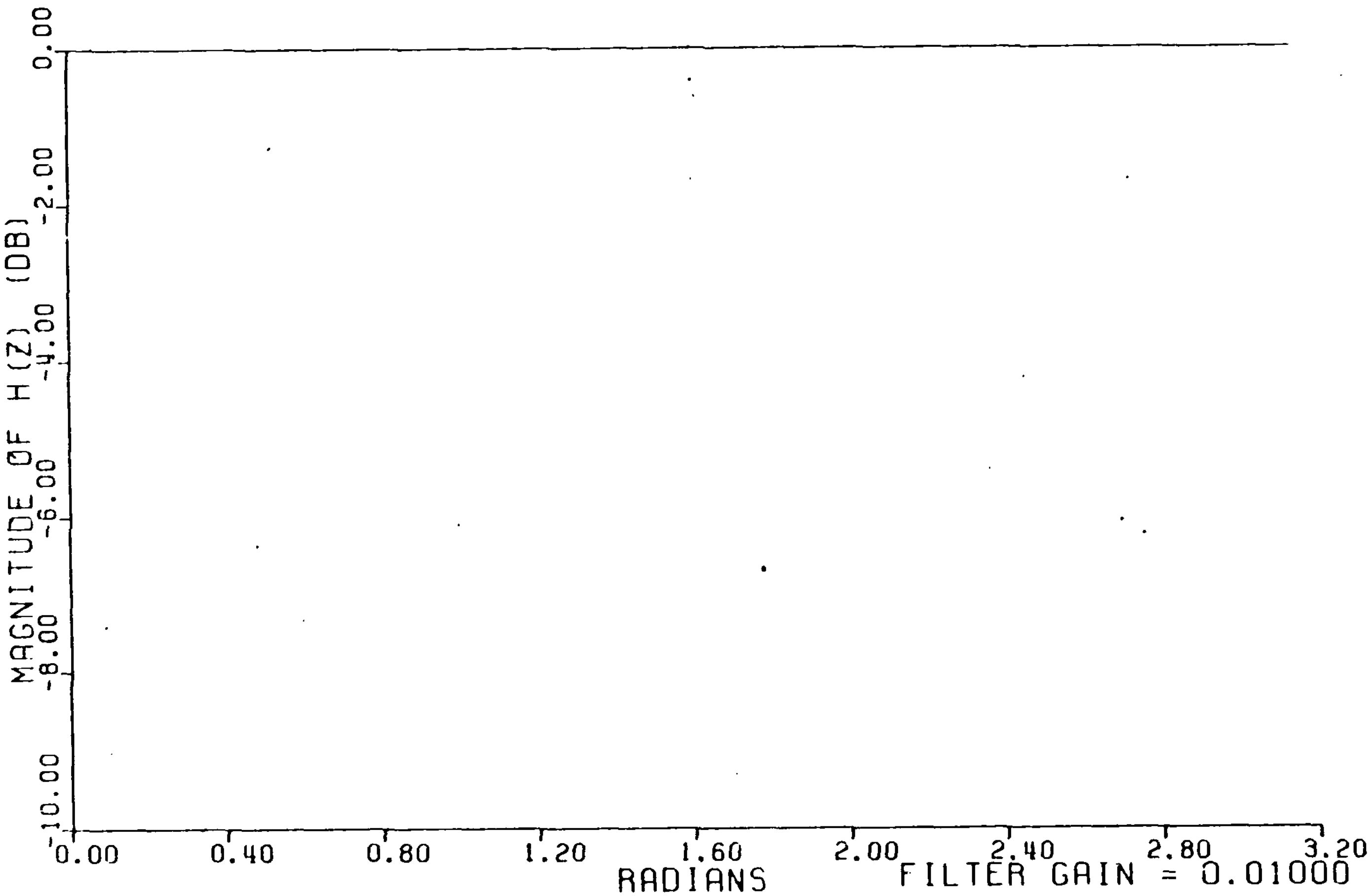


Figure 6-13g Magnitude Of H(z) In Decibels For Example Three

are close to, but outside, the unit circle. The importance of distinguishing between zeros very near the unit circle, either interior or exterior, rests on the extreme sensitivity of the phase of the frequency response as a zero moves from the interior, onto, and to the exterior of the unit circle.

VII. ADDITIONAL EXAMPLES

The purpose of this chapter is threefold. The primary objective is to provide the user with a series of plots demonstrating the general form of those types of filters not considered in chapter six. Additionally, this chapter demonstrates the responses generated by poles located at a series of positions within the unit circle. It further provides examples for user comparison when learning to interact with the z-plane.

The filters considered in the examples of chapter six include a fourth-order low-pass filter and a second-order all-pass filter. The sequence of figures 7-1a through 7-1g shows the general pole zero placements for a high-pass filter, and associated responses. Similarly, figure 7-2a through 7-2g show the responses for a band-pass filter. Figures 7-3a through 7-3g show a band-reject or notch filter.

The secondary purpose of the chapter is to show the user the general forms of time and frequency responses as the filter poles move in the z-plane. The sequence of figures 7-4 through 7-11 shows the responses for a real pole at various locations on the real axis. Figures 7-12 through 7-19 similarly show the responses of a complex pole pair moving across the z-plane. Frequency responses are not

shown for poles on, or outside of, the unit circle since the frequency response is not defined for these cases. Figures 7-17, 7-20, and 7-21 show the change in responses for a pair of complex poles which move away from the real axis.

The final intent of the chapter is to provide the user with additional figures for comparison should he need further practive in manipulating poles and zeros in the z-plane. It is important to note that the ASPF program will not allow the user to position poles on or outside the unit circ e. This limitation results in several plots which can not be reproduced by the user.

AD-A093 252 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/3
ADVANCED SIMULTION OF DIGITAL FILTERS.(U)
SEP 80 G S DOYLE

UNCLASSIFIED NL

2 of 4
AD
A093252

```
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z      REAL        Z   IMAGINARY    Z       RHO         Z      THETA      Z RT Z
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Z     1.4694920    Z     0.0        Z     1.4694920     Z     0.0         Z RZ Z
Z     0.6805066    Z     0.0        Z     0.6805066     Z     0.0         Z RZ Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
Z    **********    Z    ********** Z    **********     Z    ********** Z    Z
ZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZPZP
P     0.1697227    P +/- 0.3893862 P     0.4247673     P +/- 1.1597509 P CP P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
P    **********    P    ********** P    **********     P    ********** P    P
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP

NUMBER OF REAL ZEROES:  2
NUMBER OF COMPLEX ZERO PAIRS:  0
NUMBER OF REAL POLES:  0
NUMBER OF COMPLEX POLE PAIRS:  1
FILTER GAIN:  0.36697245
```

Figure 7-1a Pole Zero Locations For A Second-Order High-Pass Filter

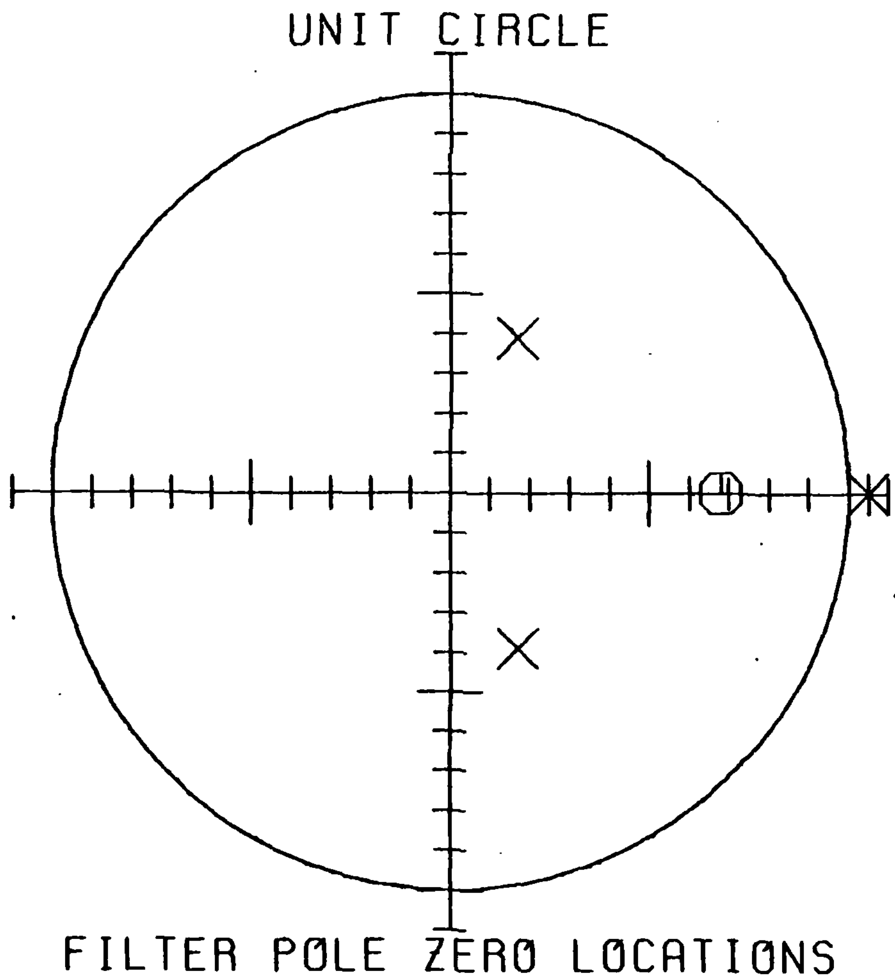


Figure 7-1b Plot Of Pole Zero Locations For A Second-Order High-Pass Filter

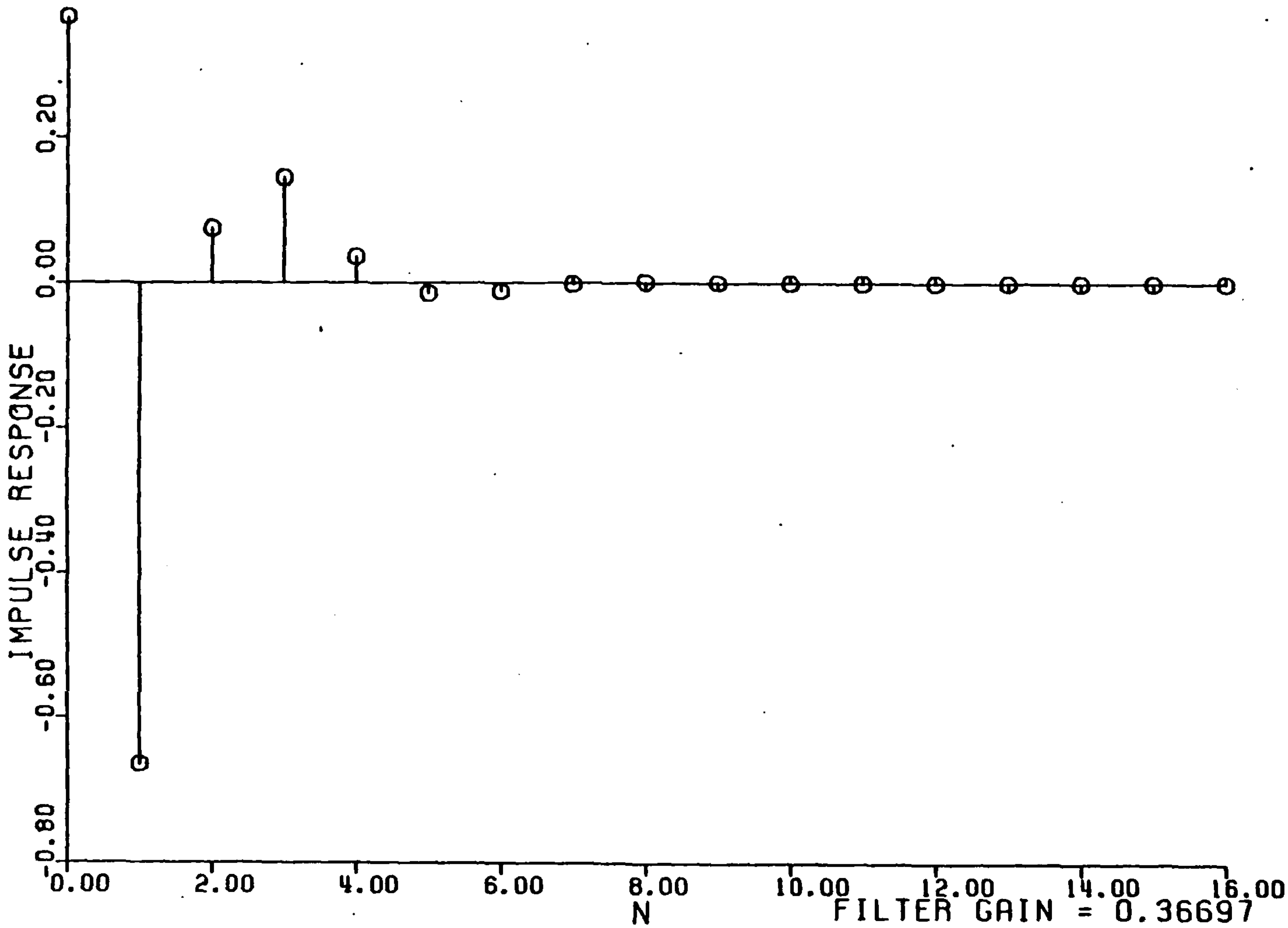


Figure 7-1c Unit Sample Response For A Second-Order High-Pass Filter

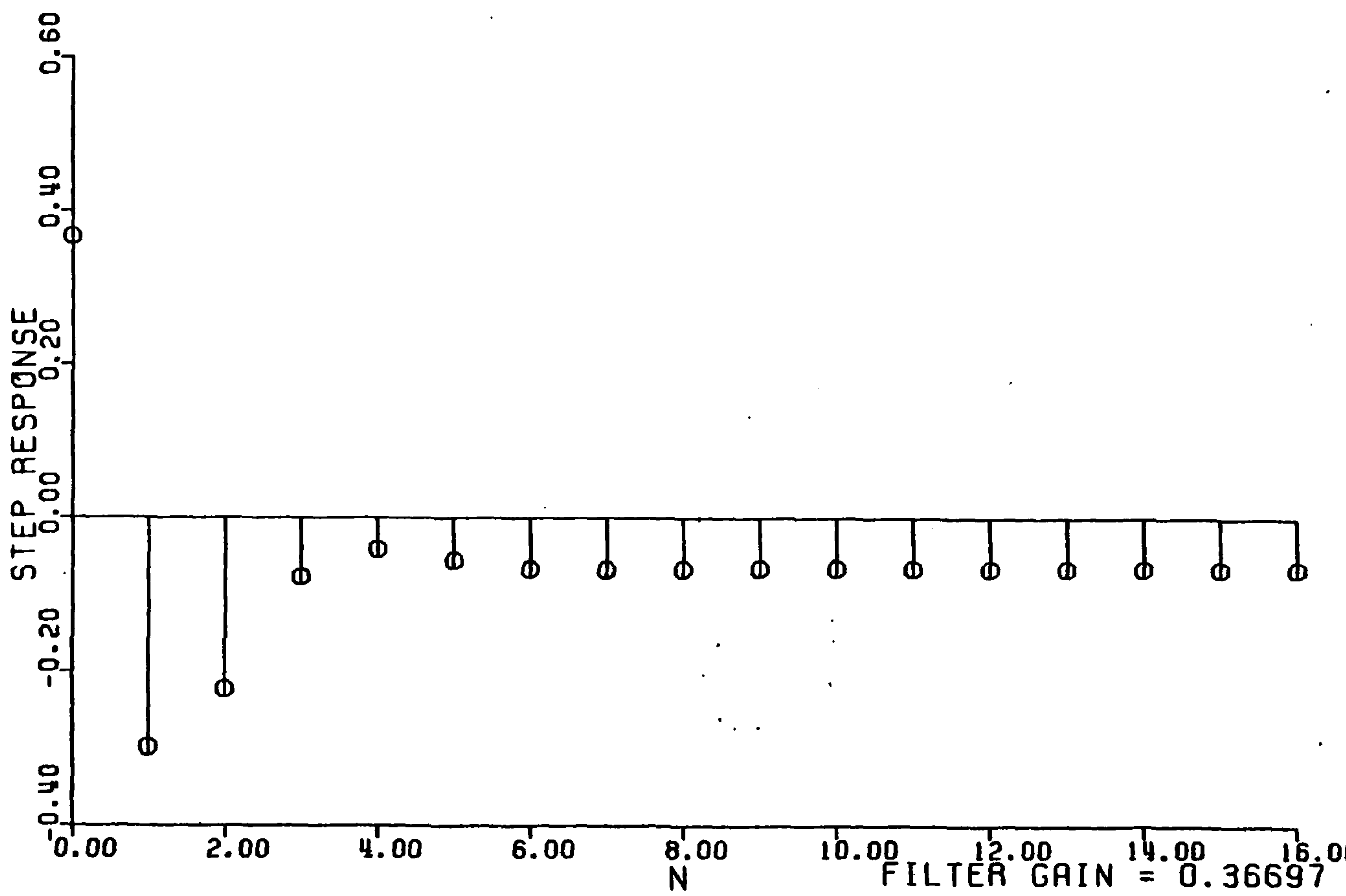


Figure 7-1d Unit Step Response For A Second-Order High-Pass Filter

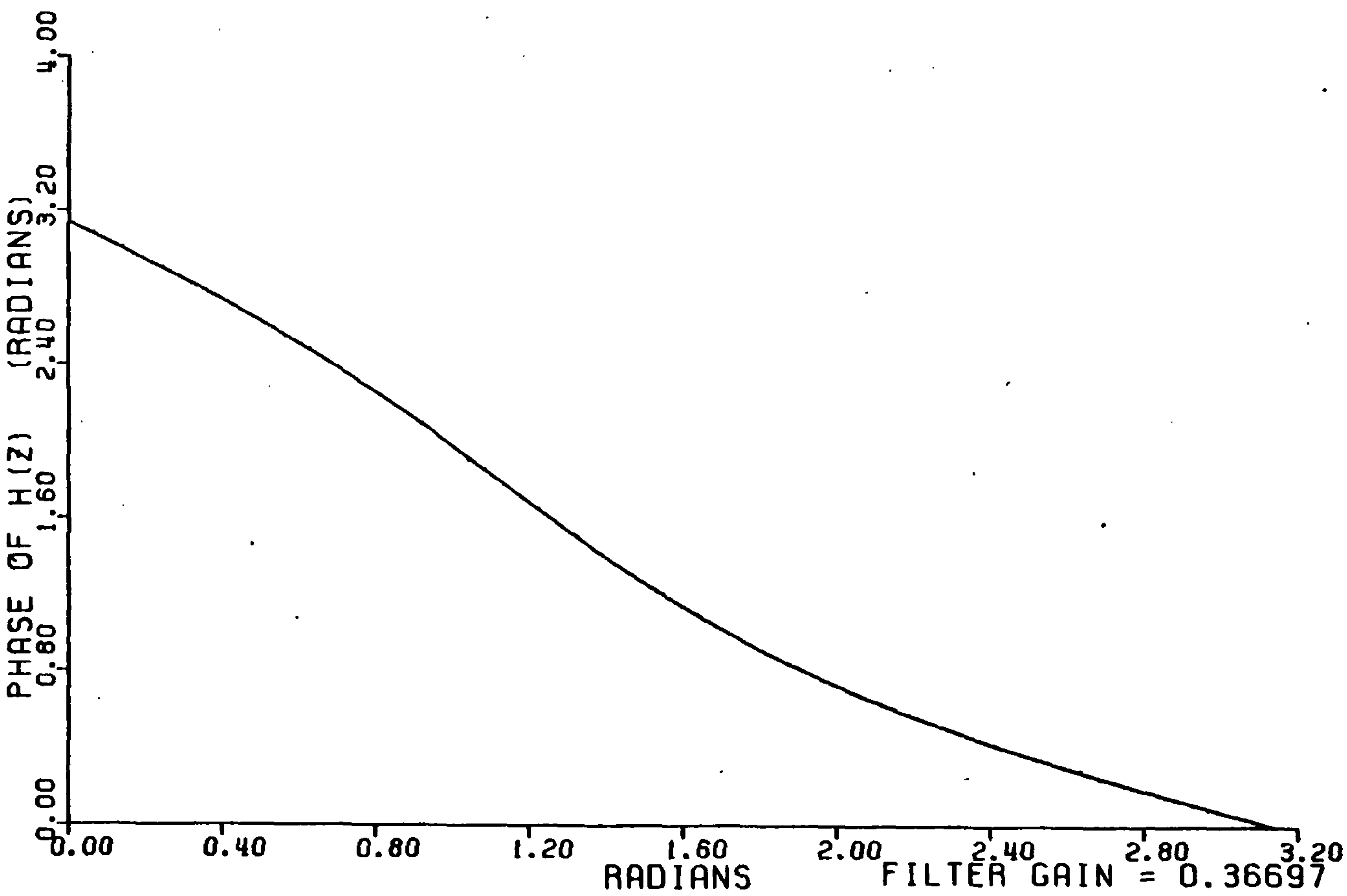


Figure 7-1e Phase Response For A Second-Order High-Pass Filter

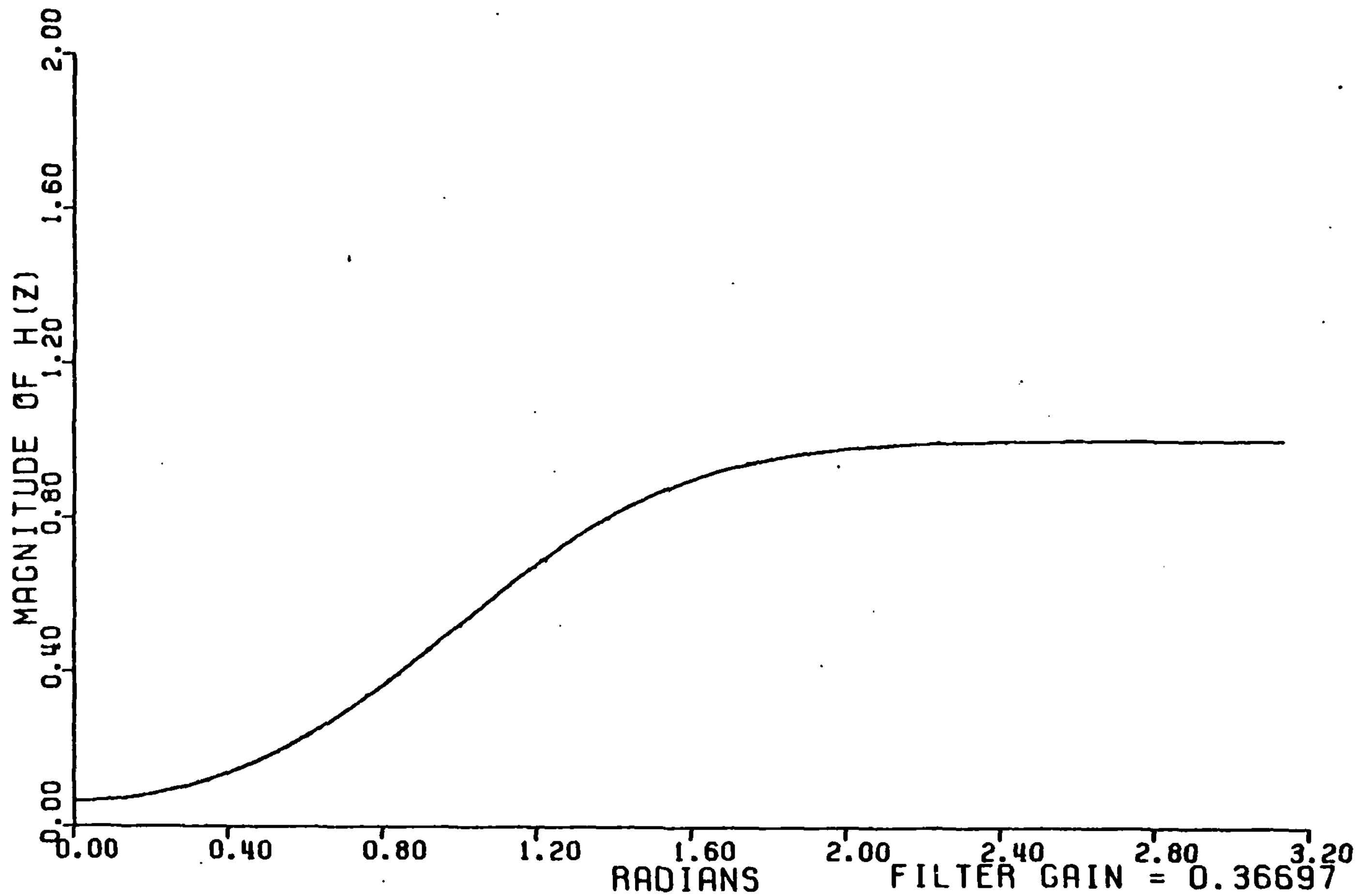


Figure 7-1f Magnitude Of H(z) For A Second-Order High-Pass Filter

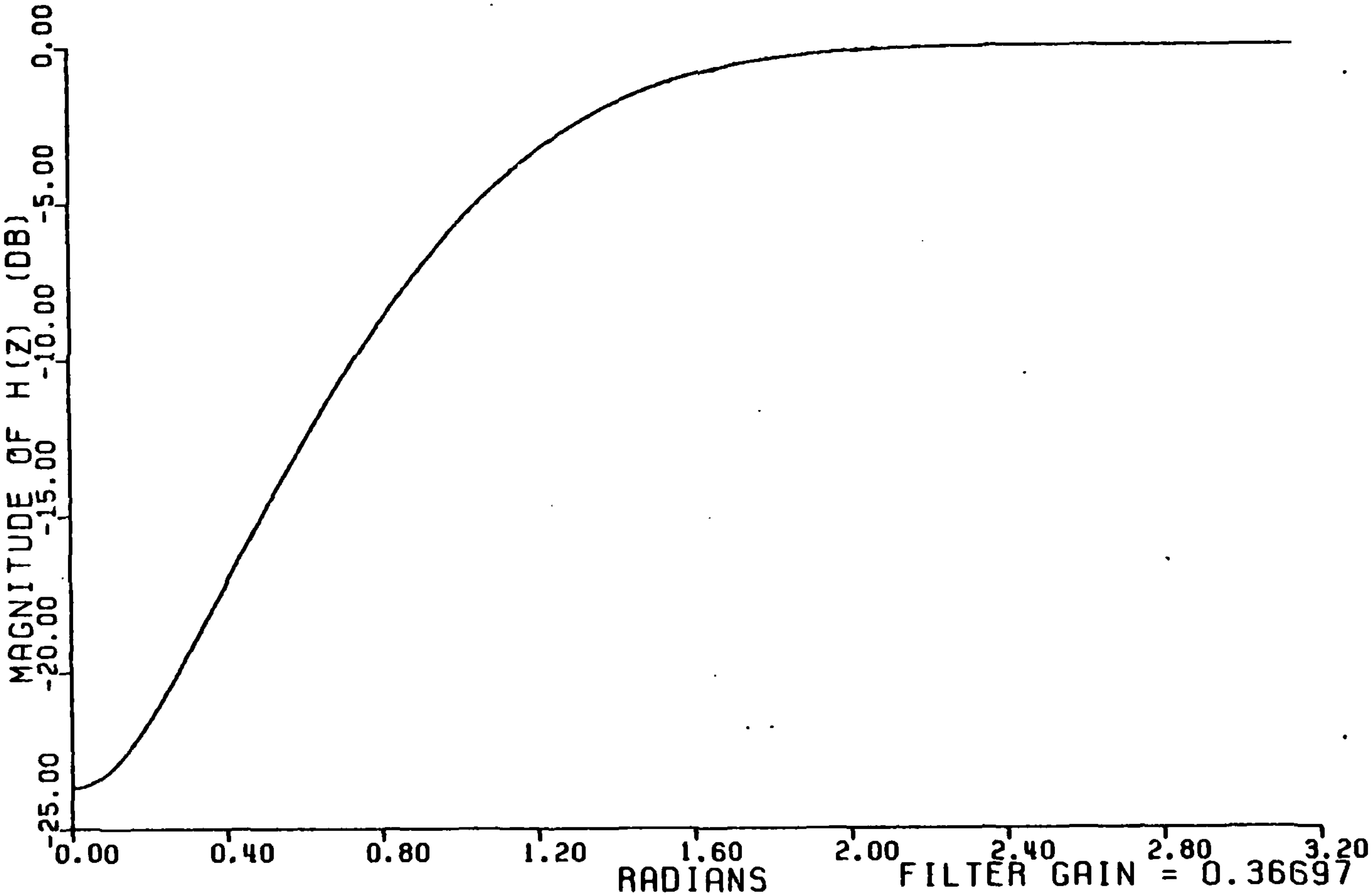


Figure 7-1g Magnitude Of H(z) In Decibels For A Second-Order High-Pass Filter

REAL	IMAGINARY	RHO	THETA	RT
0.3459291	0.0	0.3459291	0.0	RZ
-0.3459291	0.0	0.3459291	3.1415920	RZ
-2.8907642	0.0	2.8907642	3.1415920	RZ
2.8907642	0.0	2.8907642	0.0	RZ
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
0.1797063	+/- 0.7802912	0.8007175	+/- 1.3444347	CP
-0.1797063	+/- 0.7802912	0.8007175	+/- 1.7971573	CP
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	

NUMBER OF REAL ZEROES: 4
NUMBER OF COMPLEX ZERO PAIRS: 0
NUMBER OF REAL POLES: 0
NUMBER OF COMPLEX POLE PAIRS: 2
FILTER GAIN: 0.02473500

Figure 7-2a Pole Zero Locations For A Fourth-Order Band-Pass Filter

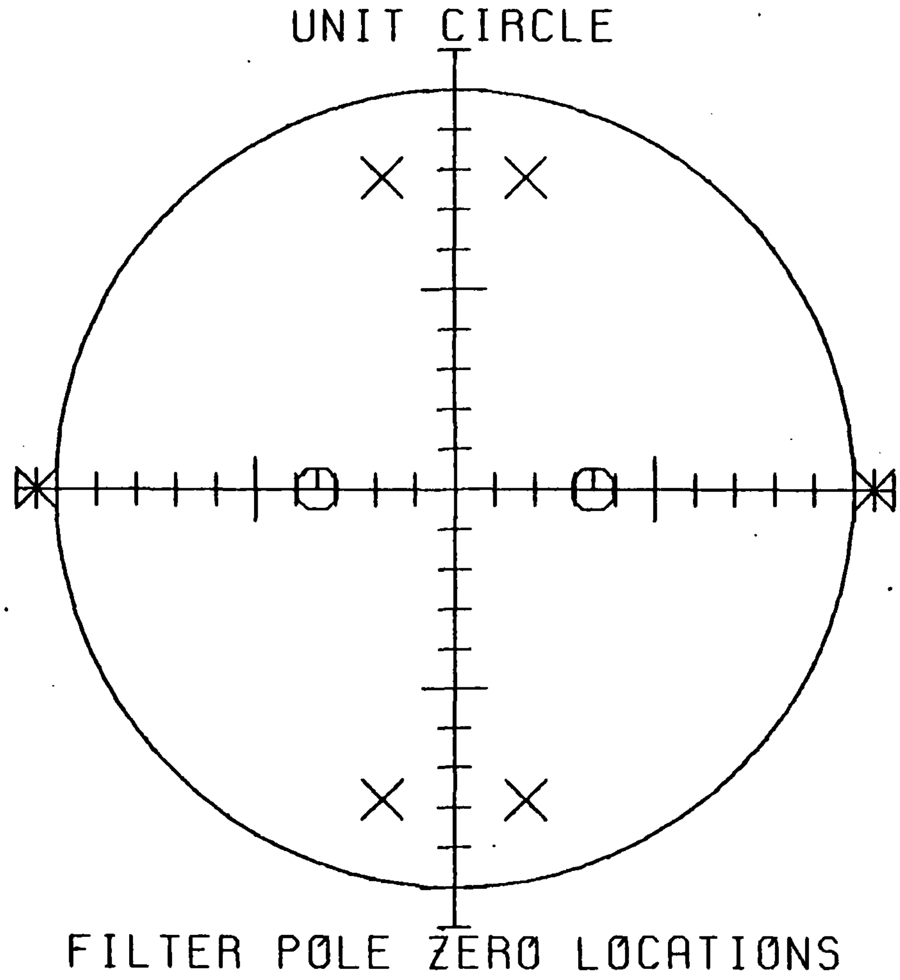


Figure 7-2b Plot Of Pole Zero Locations For A Fourth-Order Band-Pass Filter

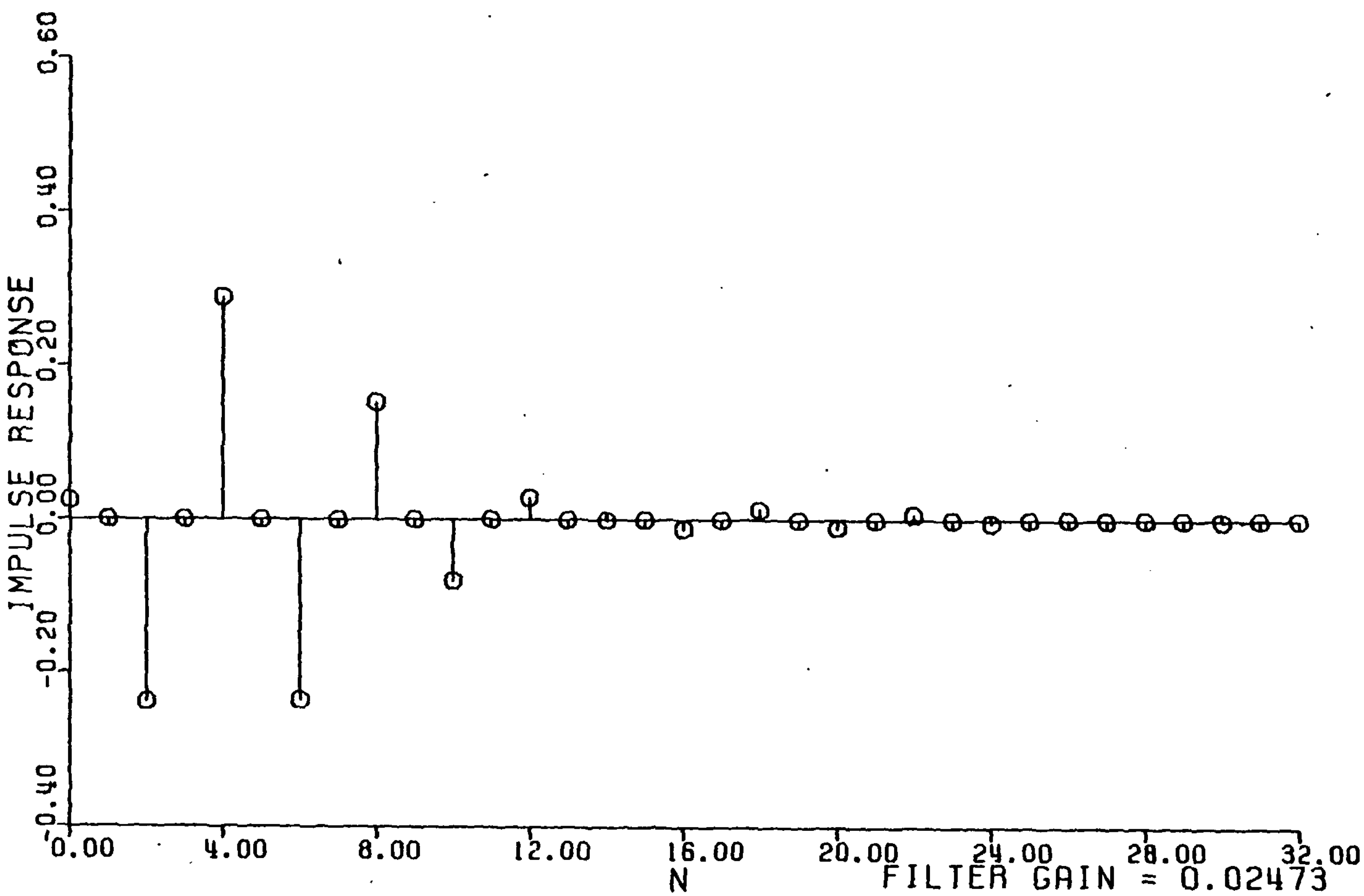


Figure 7-2c Unit Sample Response For A Fourth-Order Band-Pass Filter

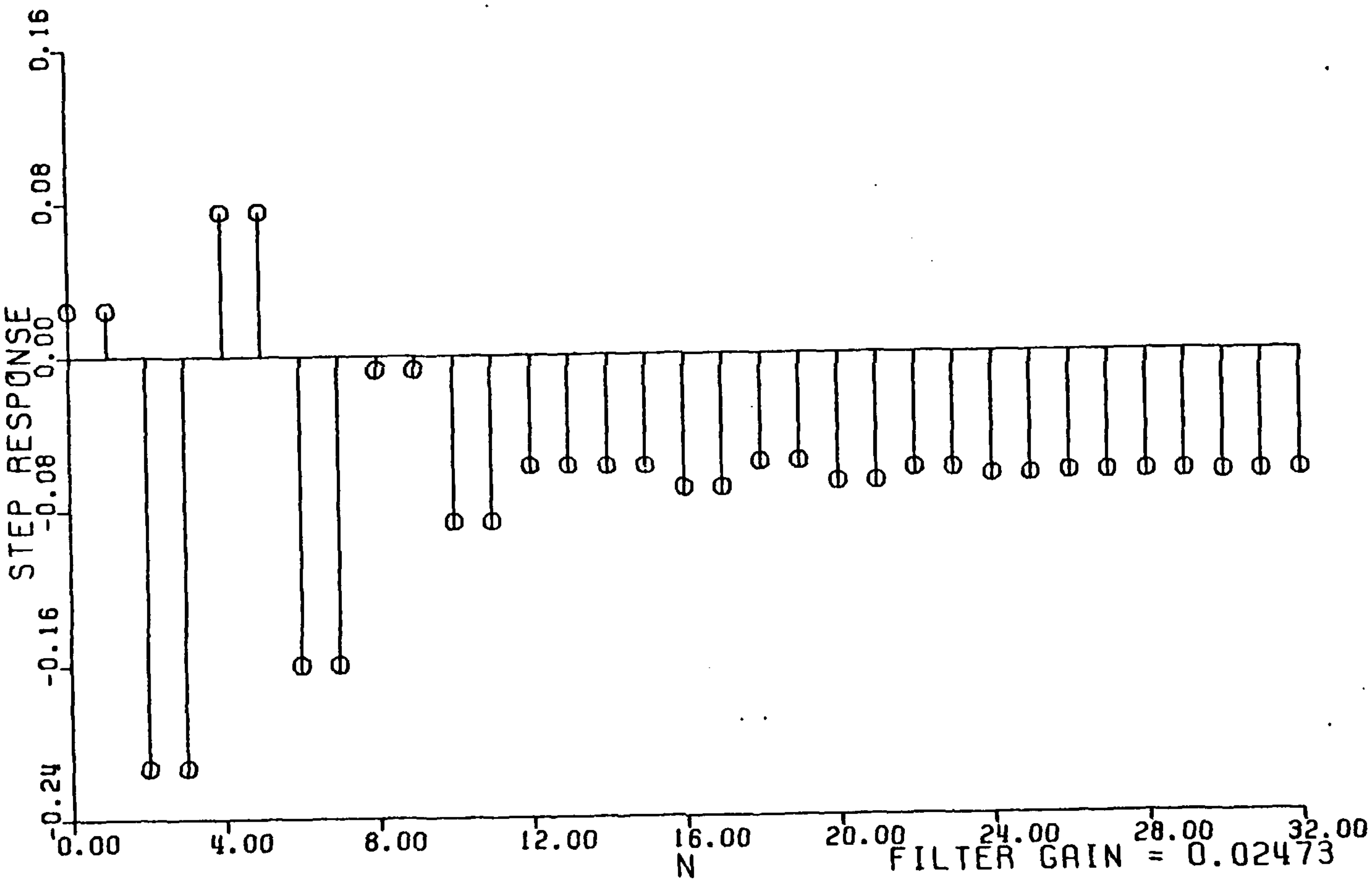


Figure 7-2d Unit Step Response For A Fourth-Order Band-Pass Filter

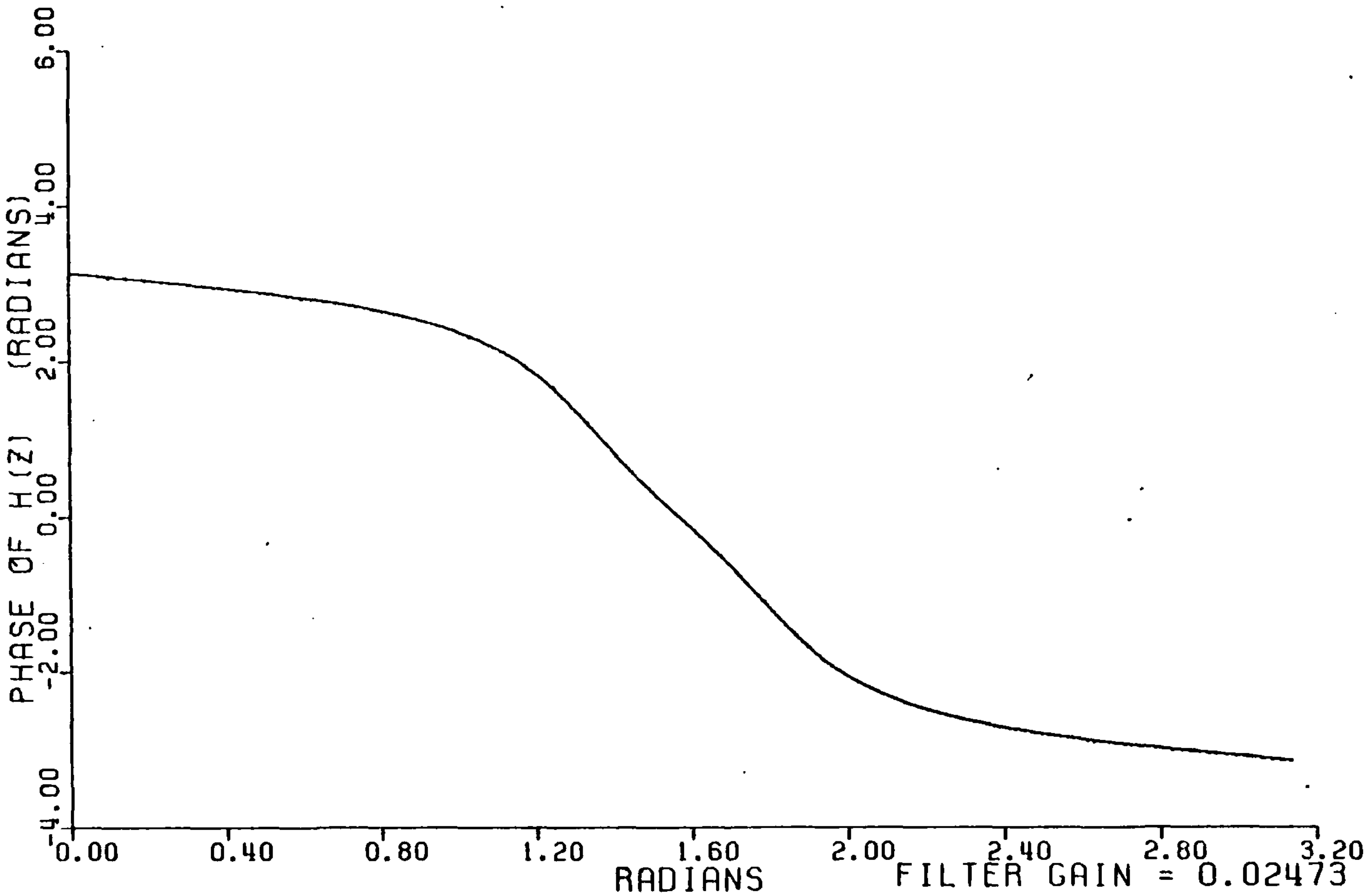


Figure 7-2e Phase Response For A Fourth-Order Band-Pass Filter

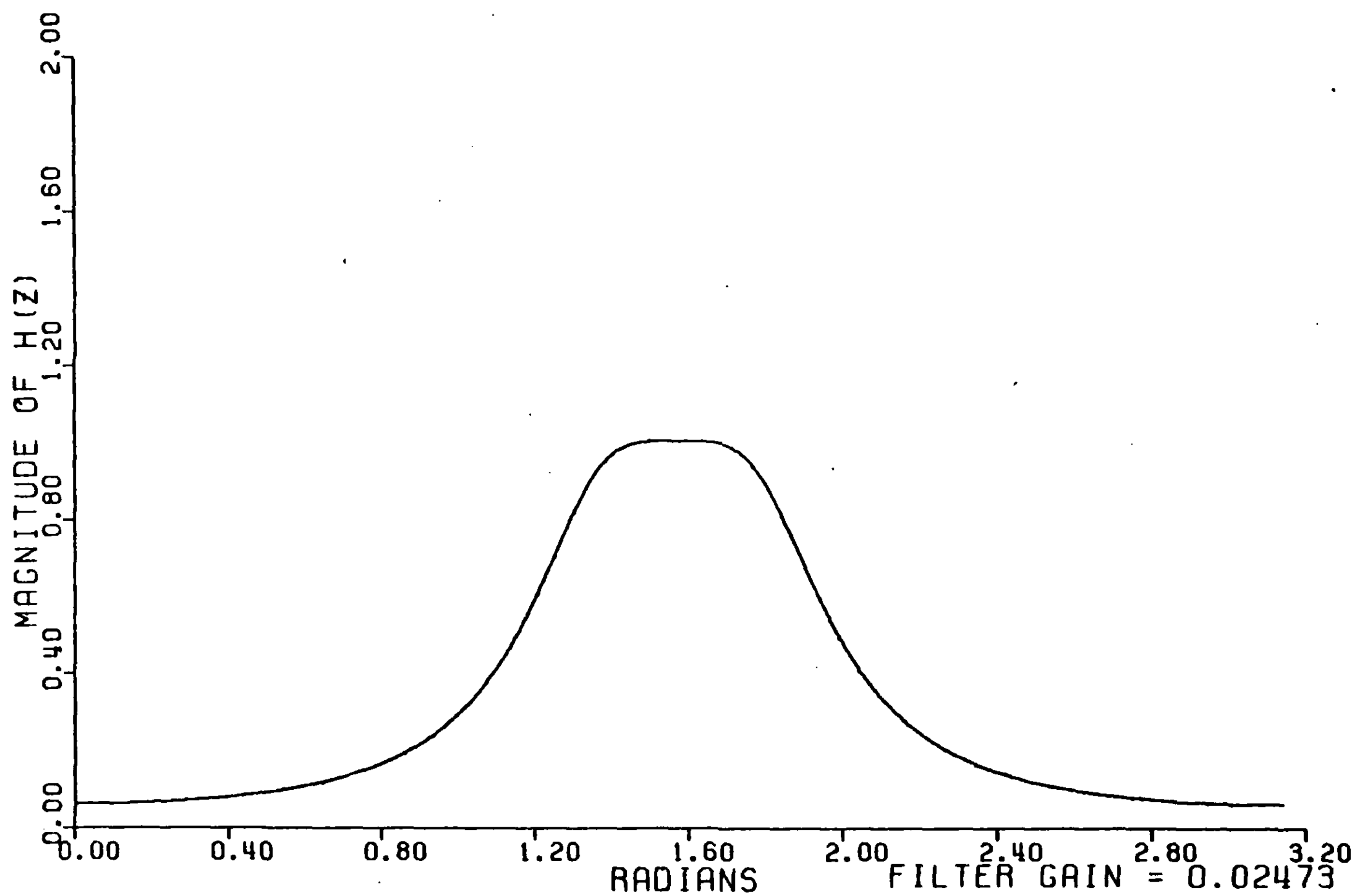


Figure 7-2f Magnitude Of H(z) For a Fourth-Order Band-Pass Filter

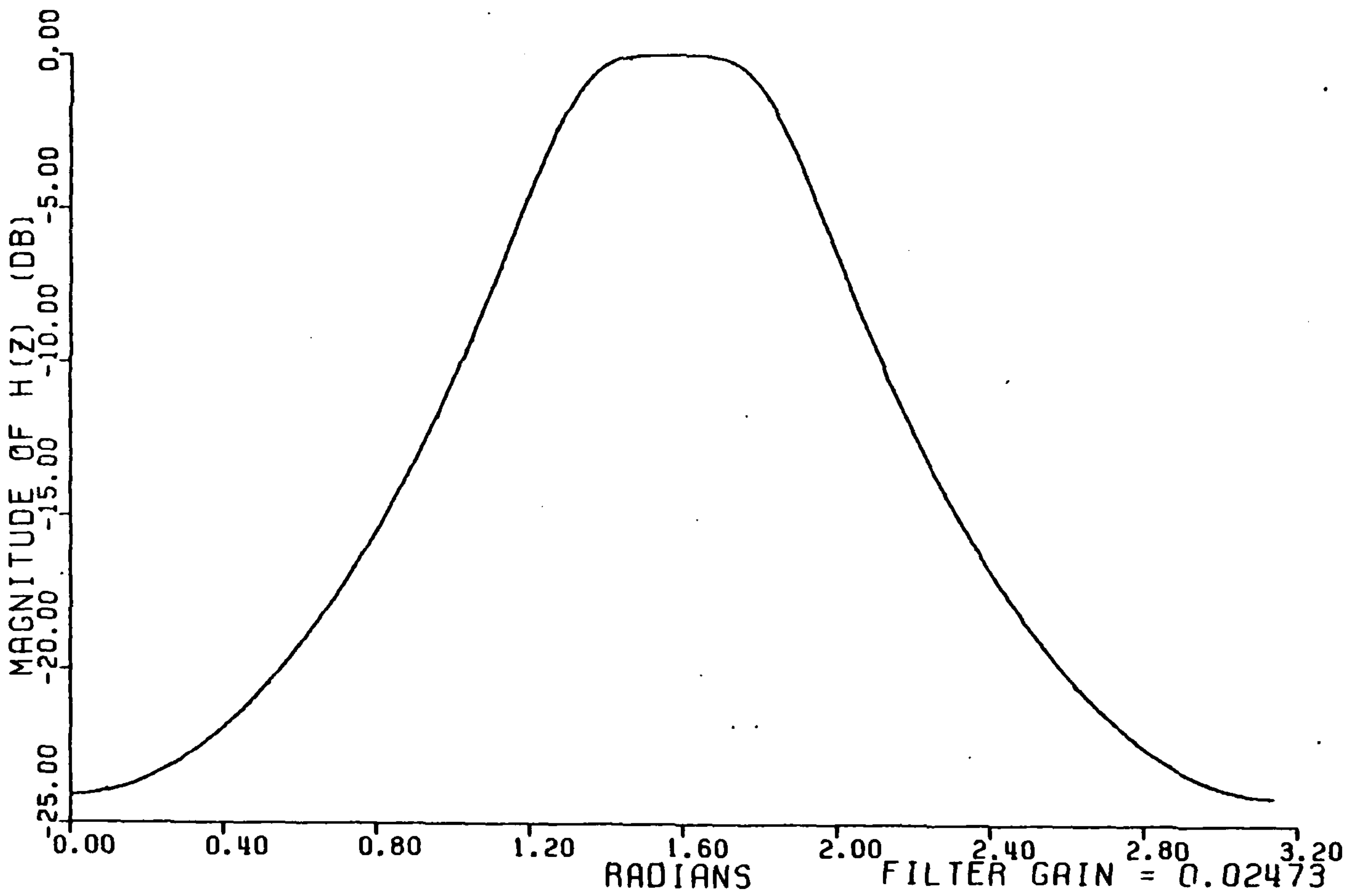


Figure 7-2g Magnitude Of H(z) For A Fourth-Order Band-Pass Filter

REAL	IMAGINARY	RHO	THETA	RT
0.0	+/- 0.9211732	0.9211732	+/- 1.5707951	CZ
0.0	+/- 1.0855713	1.0855713	+/- 1.5707951	CZ
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
0.1852023	+/- 0.7716743	0.7935876	+/- 1.3352489	CP
-0.1852023	+/- 0.7716743	0.7935876	+/- 1.8063431	CP
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	
**********	**********	**********	**********	

NUMBER OF REAL ZEROES: 0
NUMBER OF COMPLEX ZERO PAIRS; 2
NUMBER OF REAL POLES: 0
NUMBER OF COMPLEX POLE PAIRS: 2
FILTER GAIN: 0.62447256

Figure 7-3a Pole Zero Locations For A Fourth-Order Notch Filter

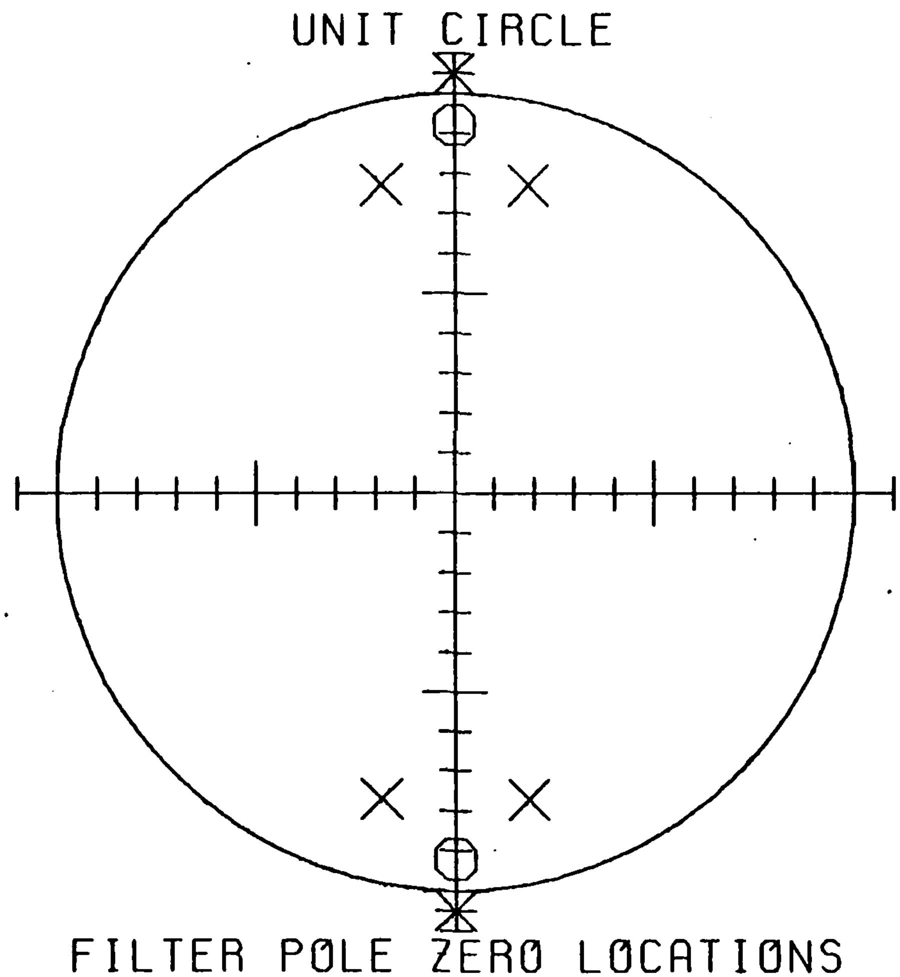


Figure 7-3b Plot Of Pole Zero Locations For A Fourth-Order Notch Filter

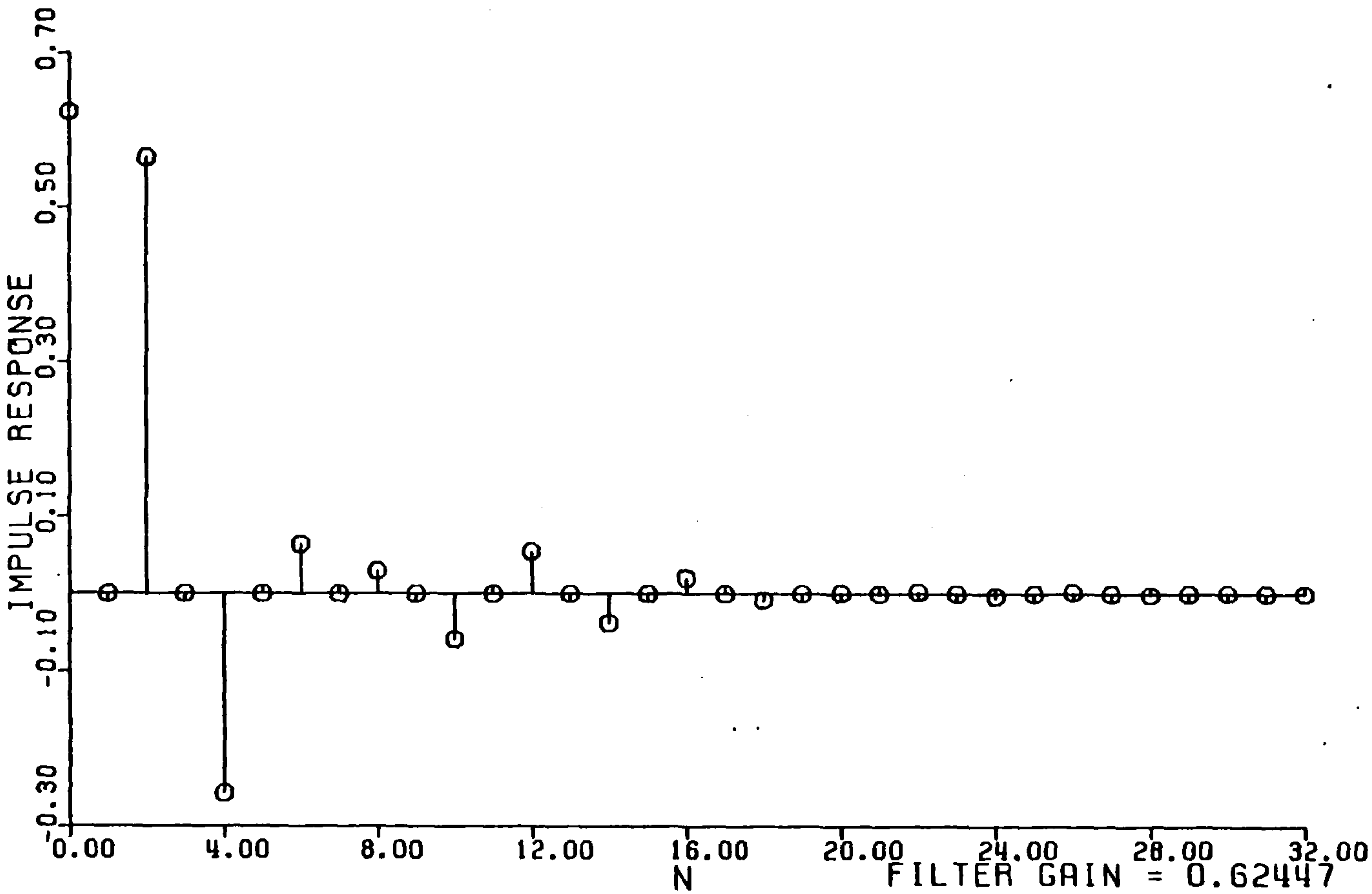


Figure 7-3c Unit Sample Response For A Fourth-Order Notch Filter

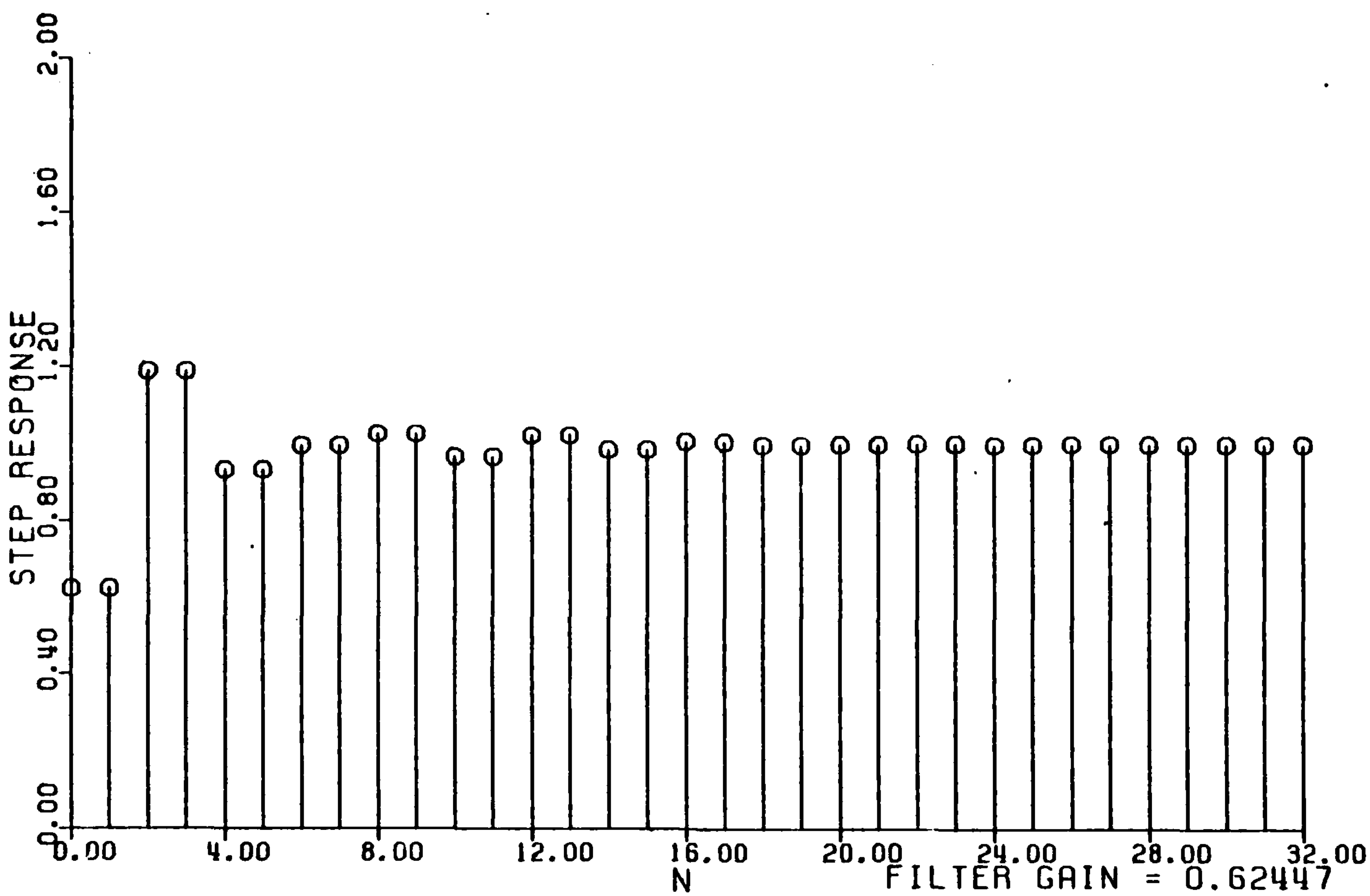


Figure 7-3d Unit Step Response For A Fourth-Order Notch Filter

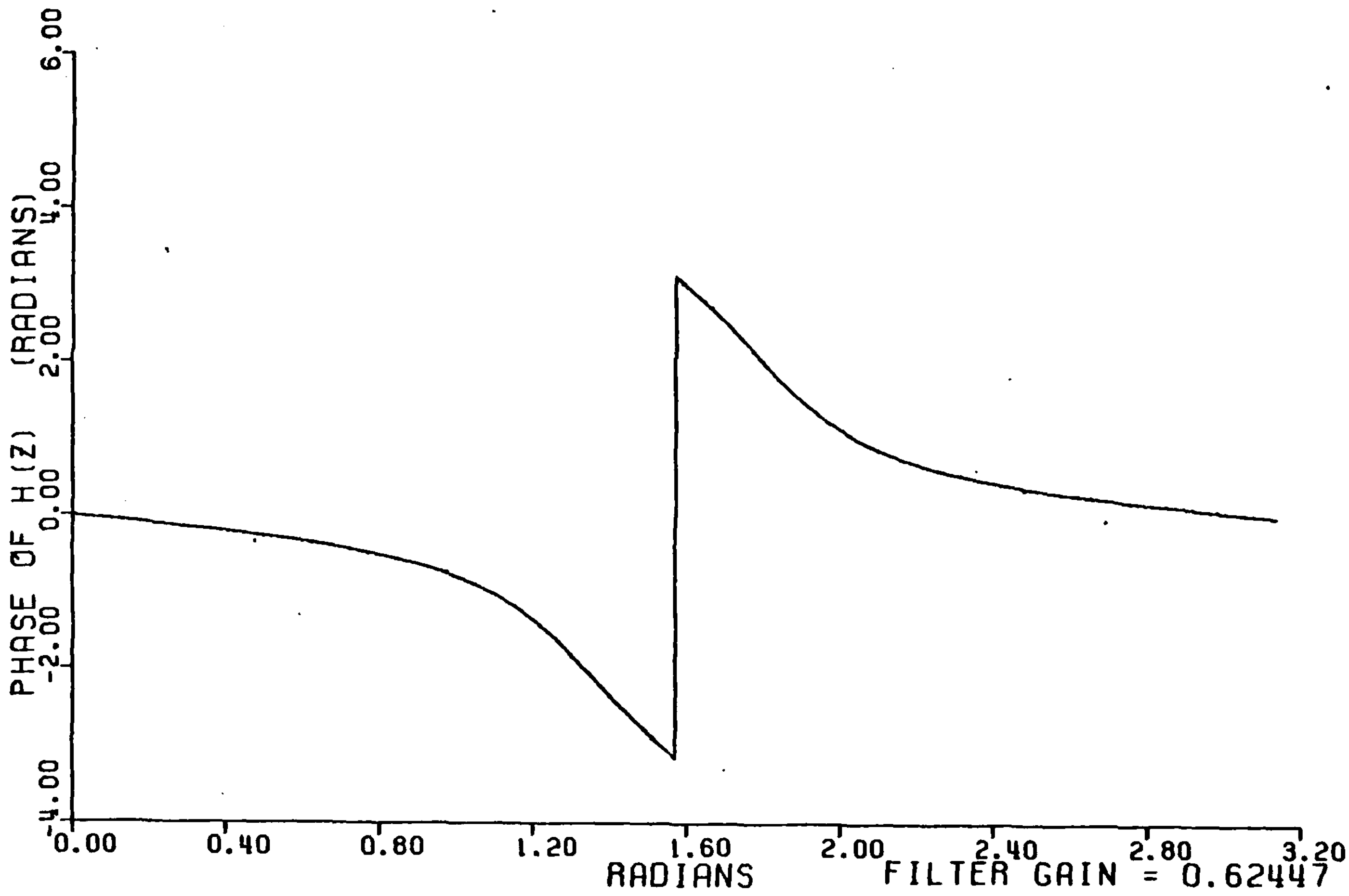


Figure 7-3e Phase Response For a Fourth-Order Notch Filter

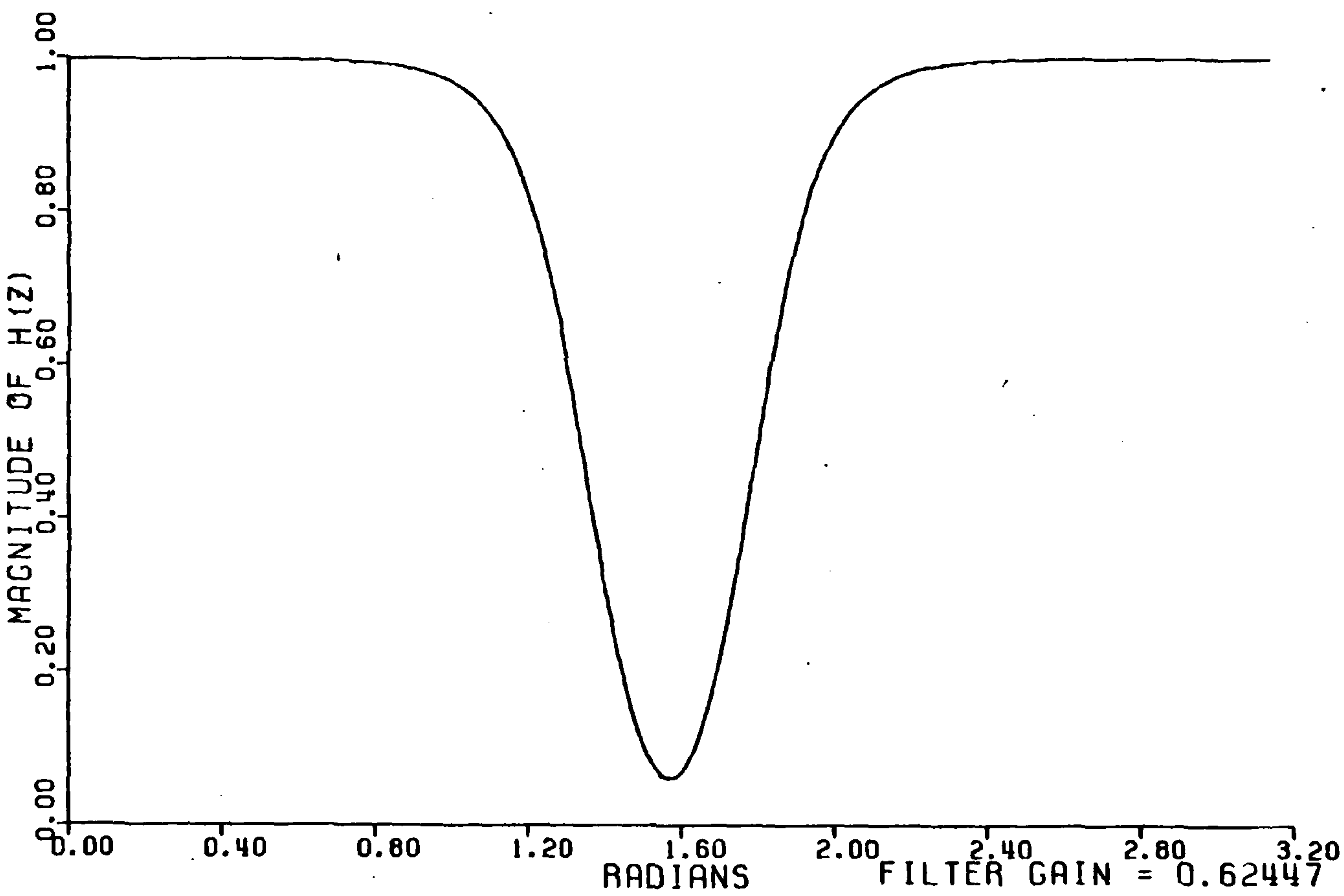


Figure 7-3f Magnitude Of H(z) For A Fourth-Order Notch Filter

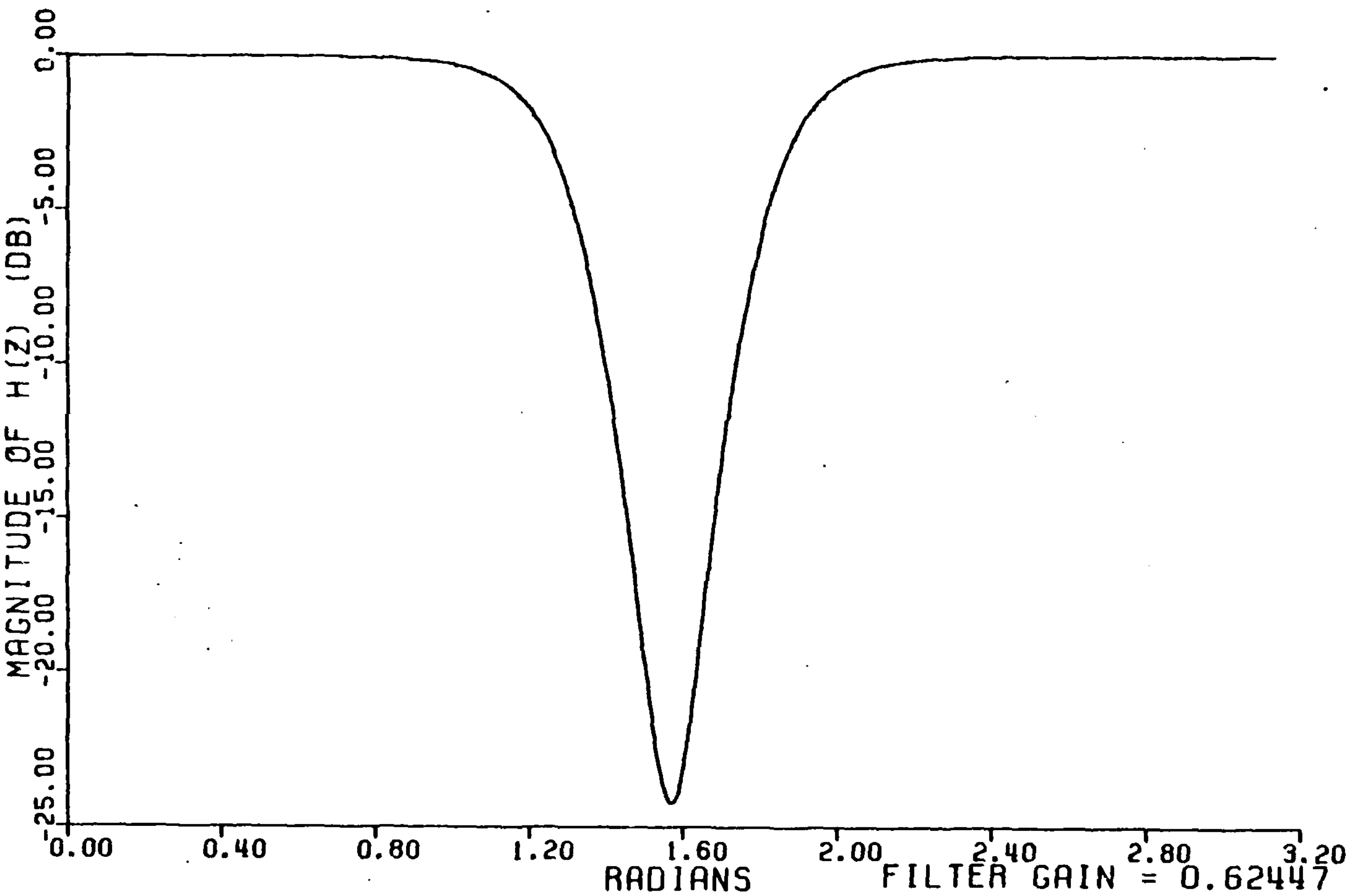


Figure 7-3g Magnitude In Decibels For A Fourth-Order Notch Filter

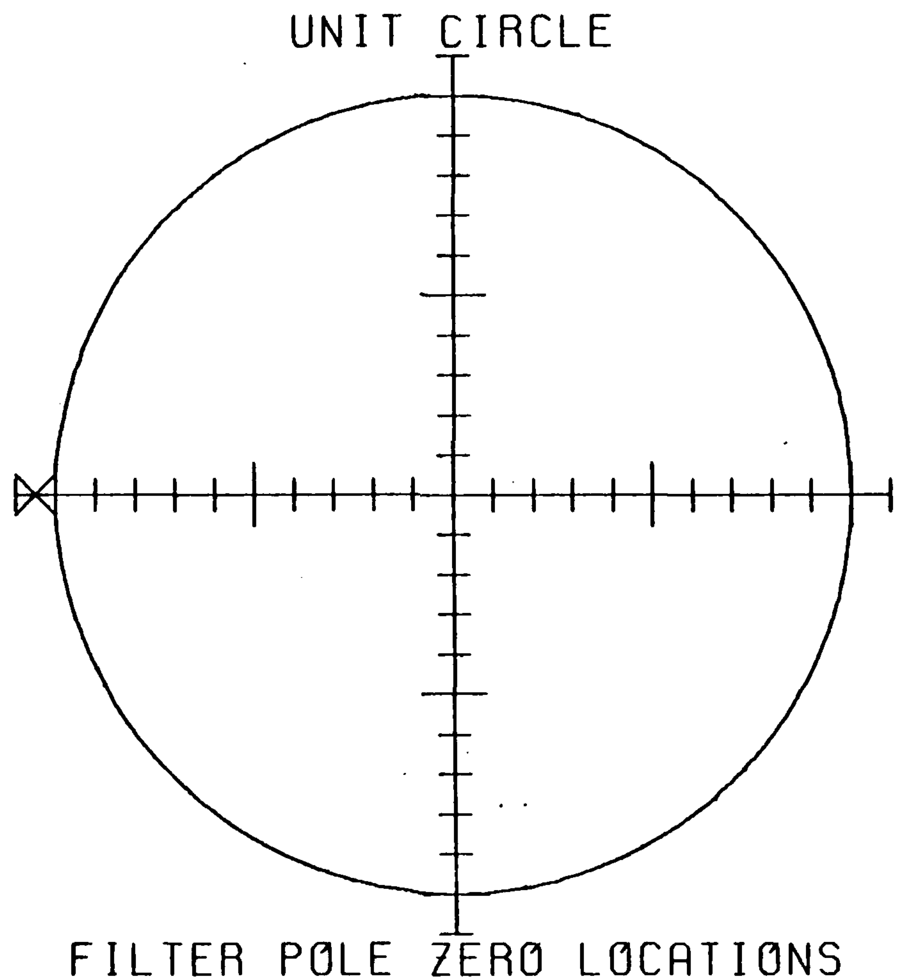


Figure 7-4a Example Of A Single Real Pole Outside The Unit Circle At (-1.05)

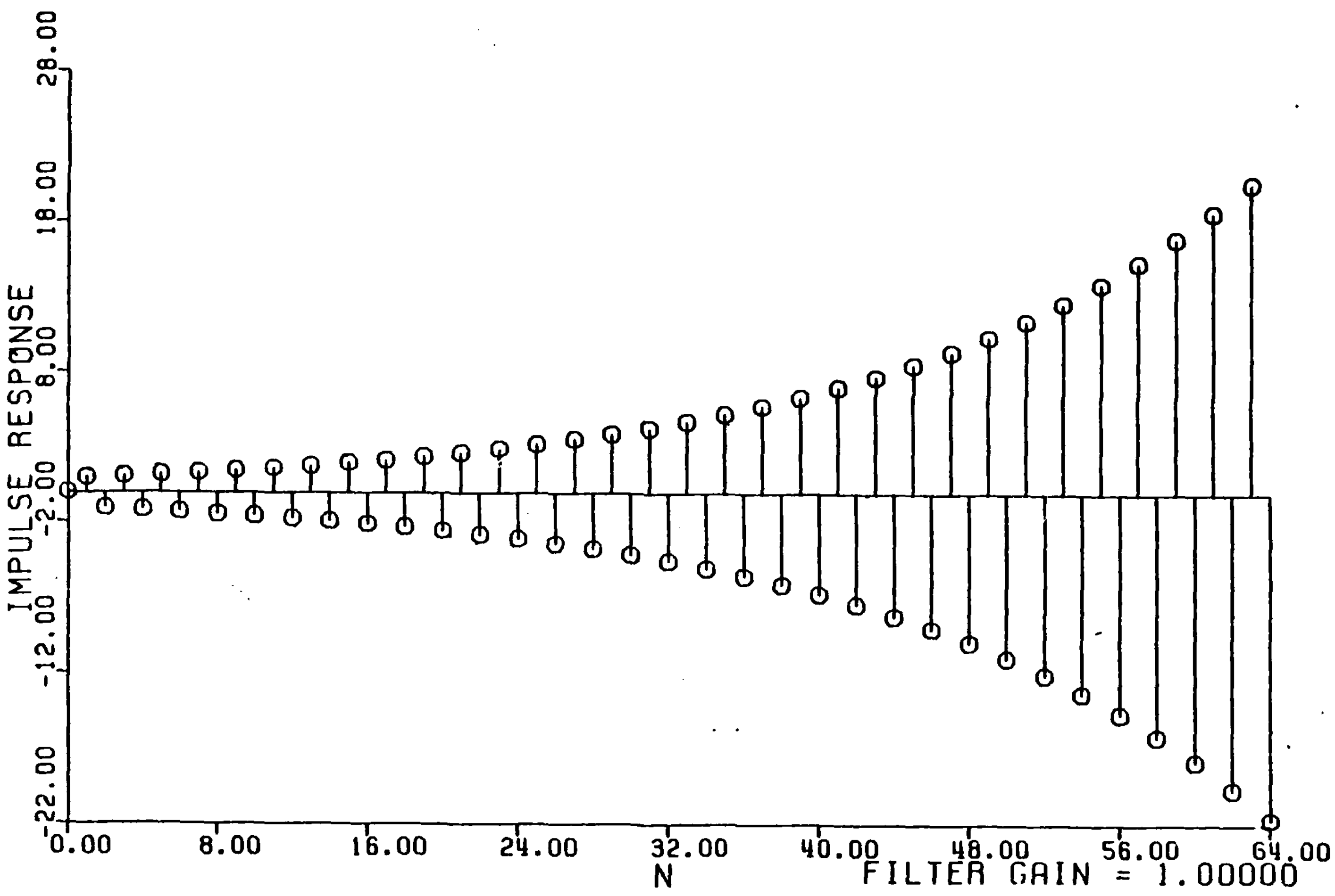


Figure 7-4b Unit Sample Response For A Single Real Pole At (-1.05)

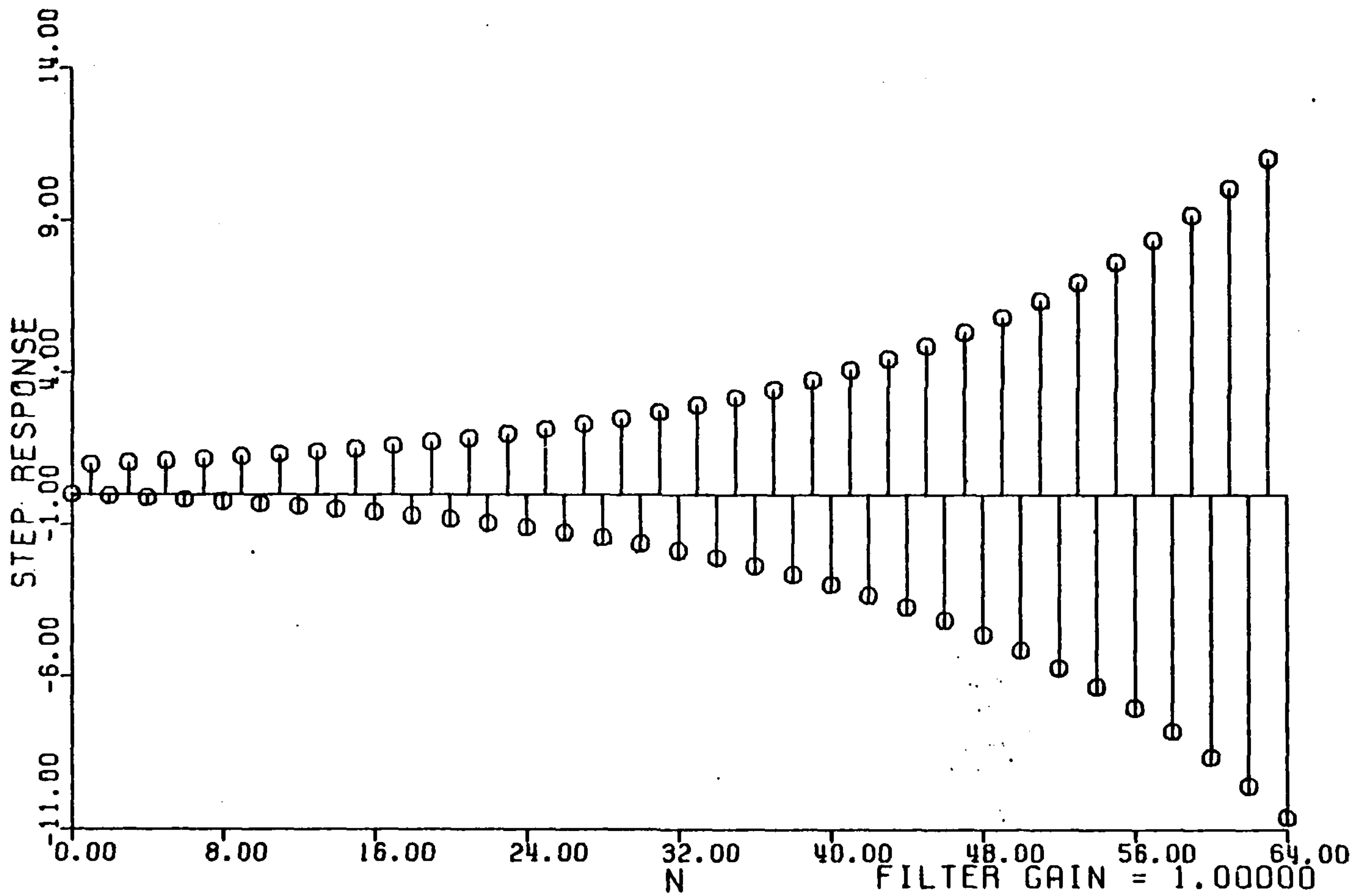


Figure 7-4c Unit Step Response For A Single Real Pole At (-1.05)

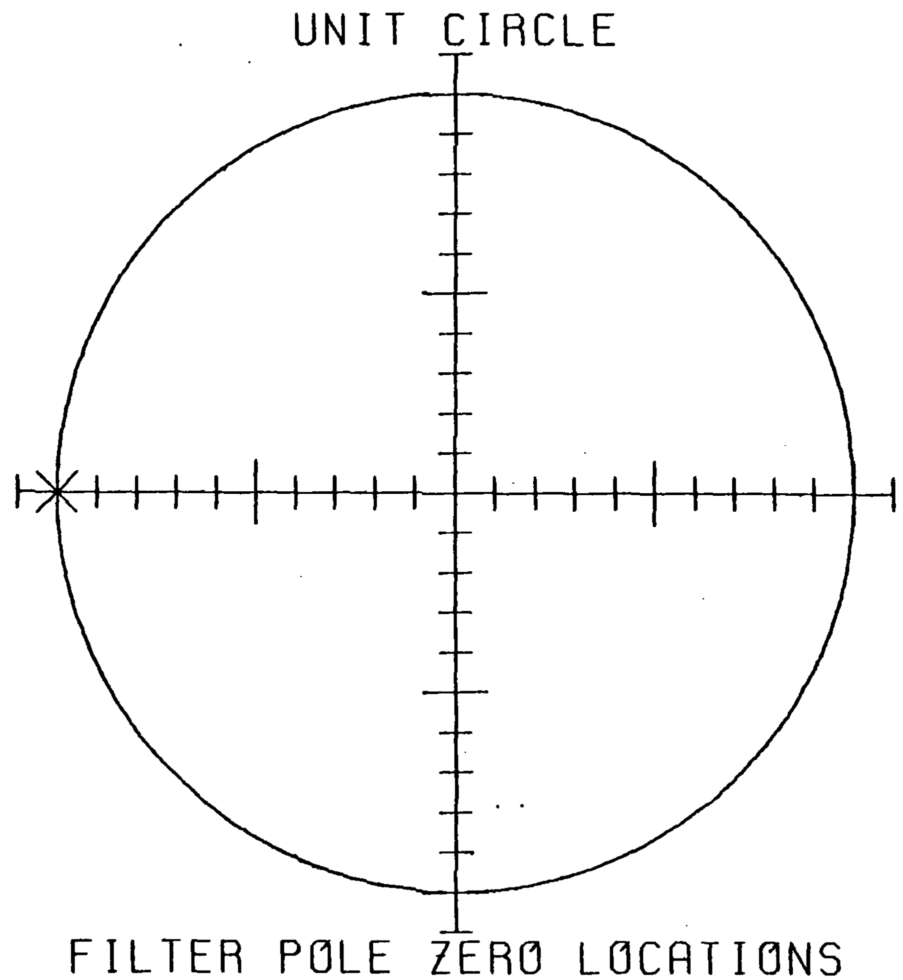


Figure 7-5a Example Of A Single Real Pole On The Unit Circle At (-1.0)

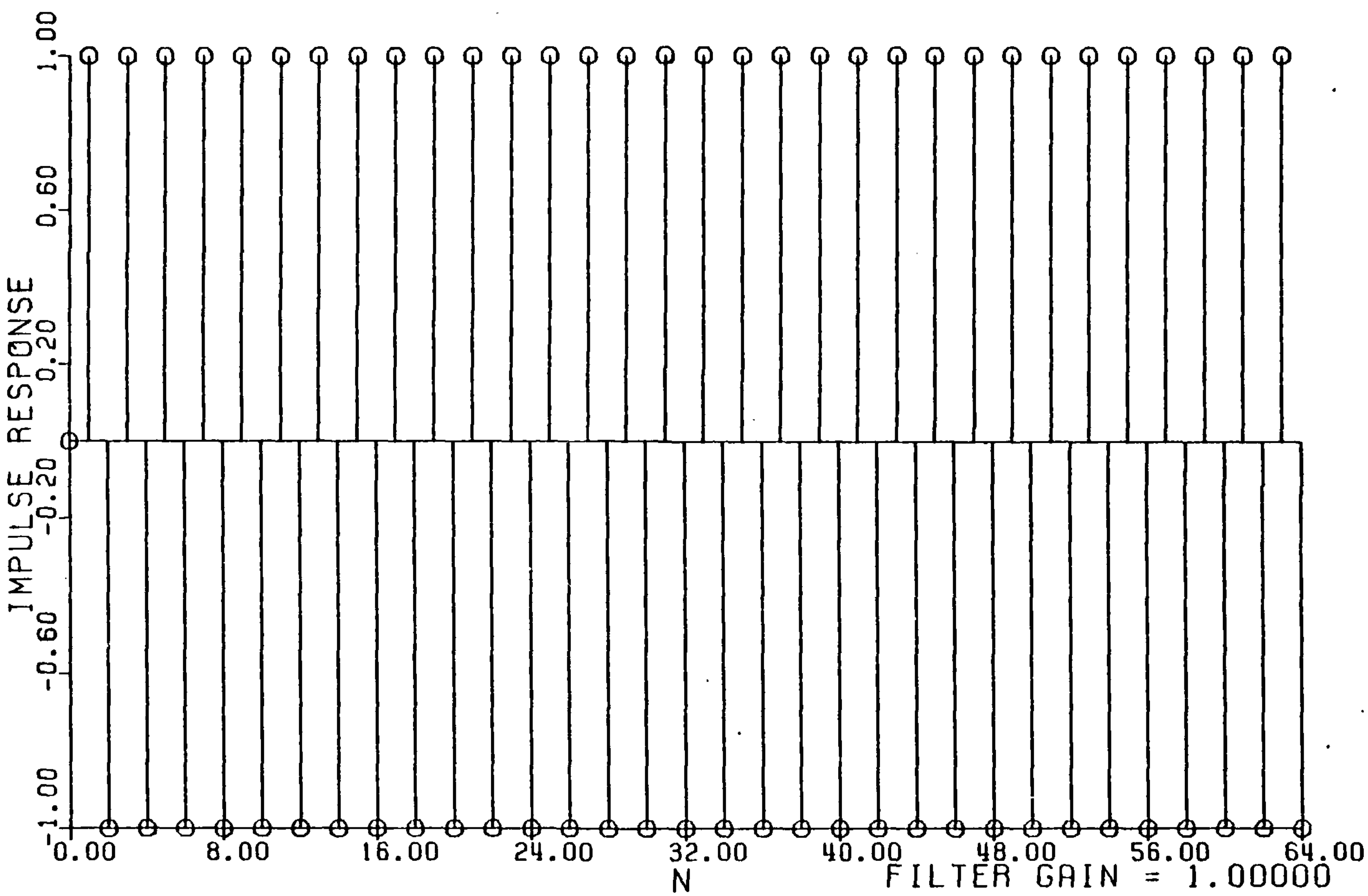


Figure 7-5b Unit Sample Response For A Single Real Pole At (-1.0)

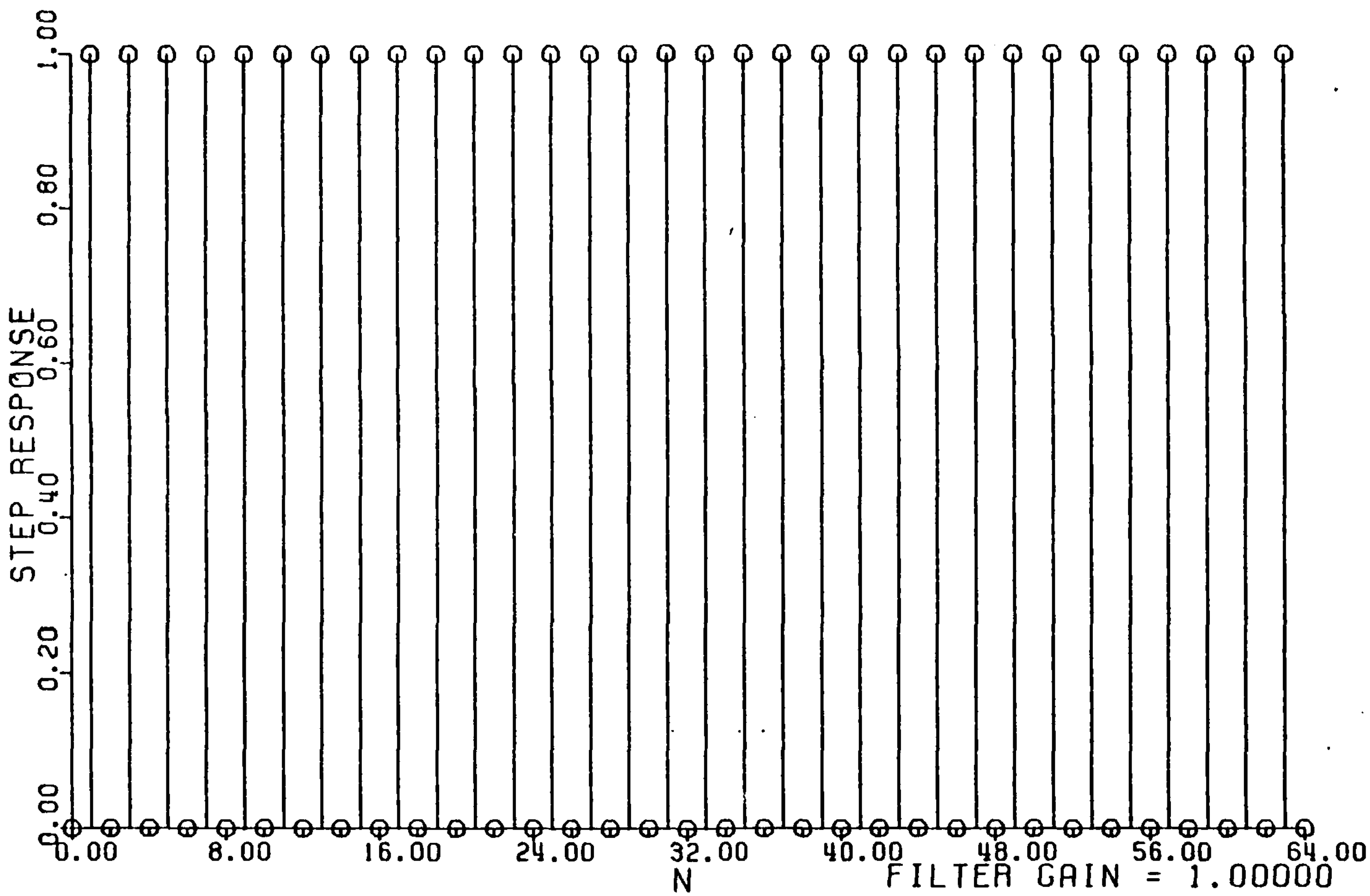


Figure 7-5c Unit Step Response For A Single Real Pole At (-1.0)

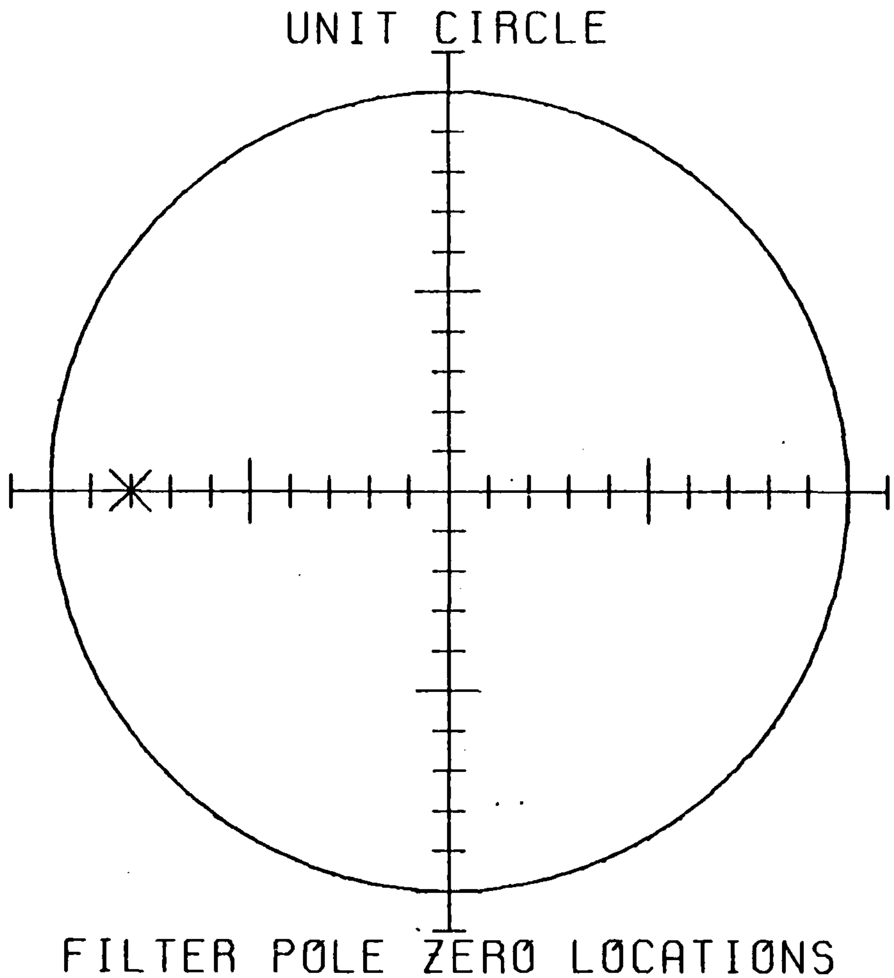


Figure 7-6a Example Of A Single Real Pole At (-0.8)

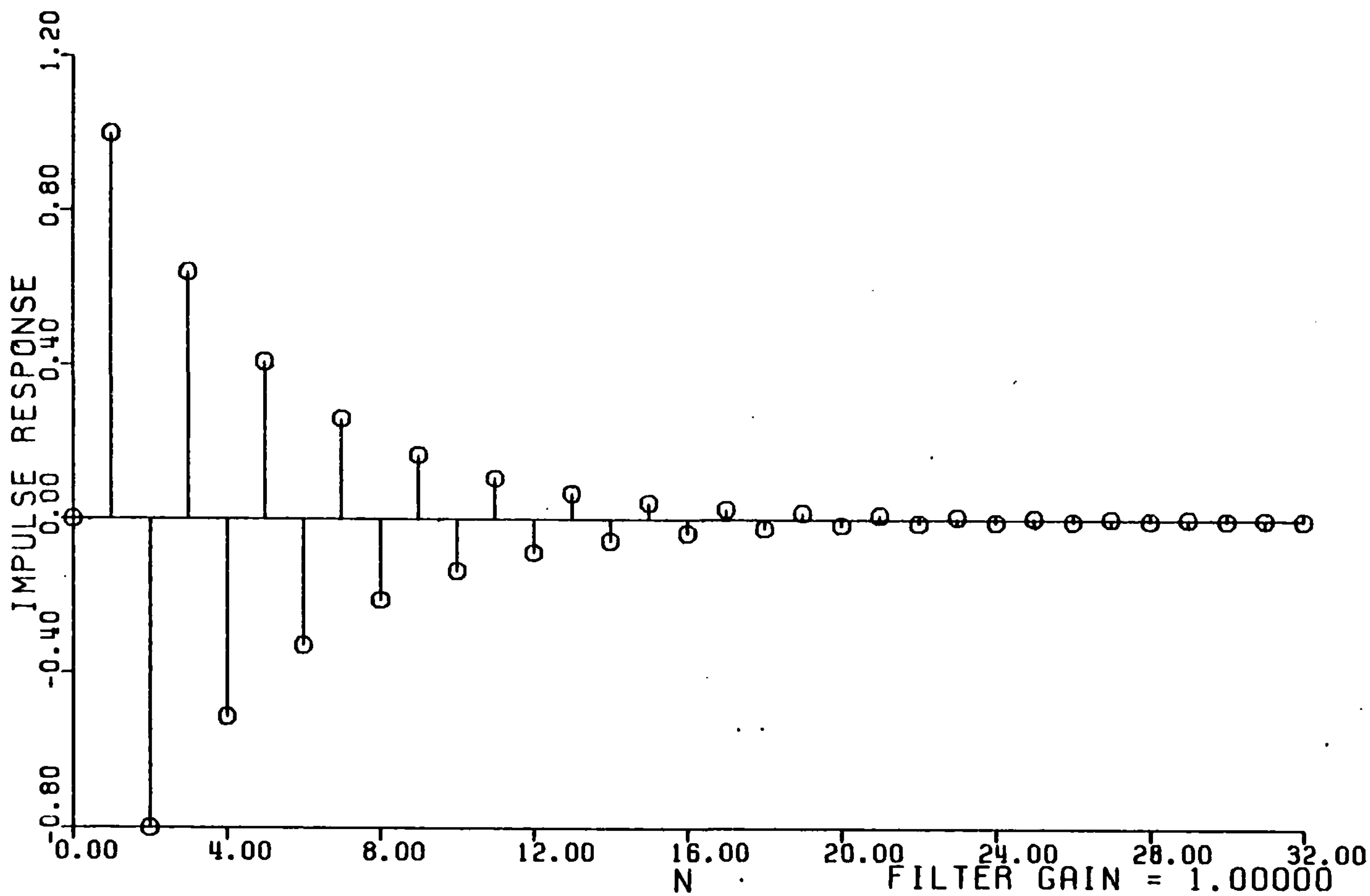


Figure 7-6b Unit Sample Response For A Single Real Pole at (-1.0)

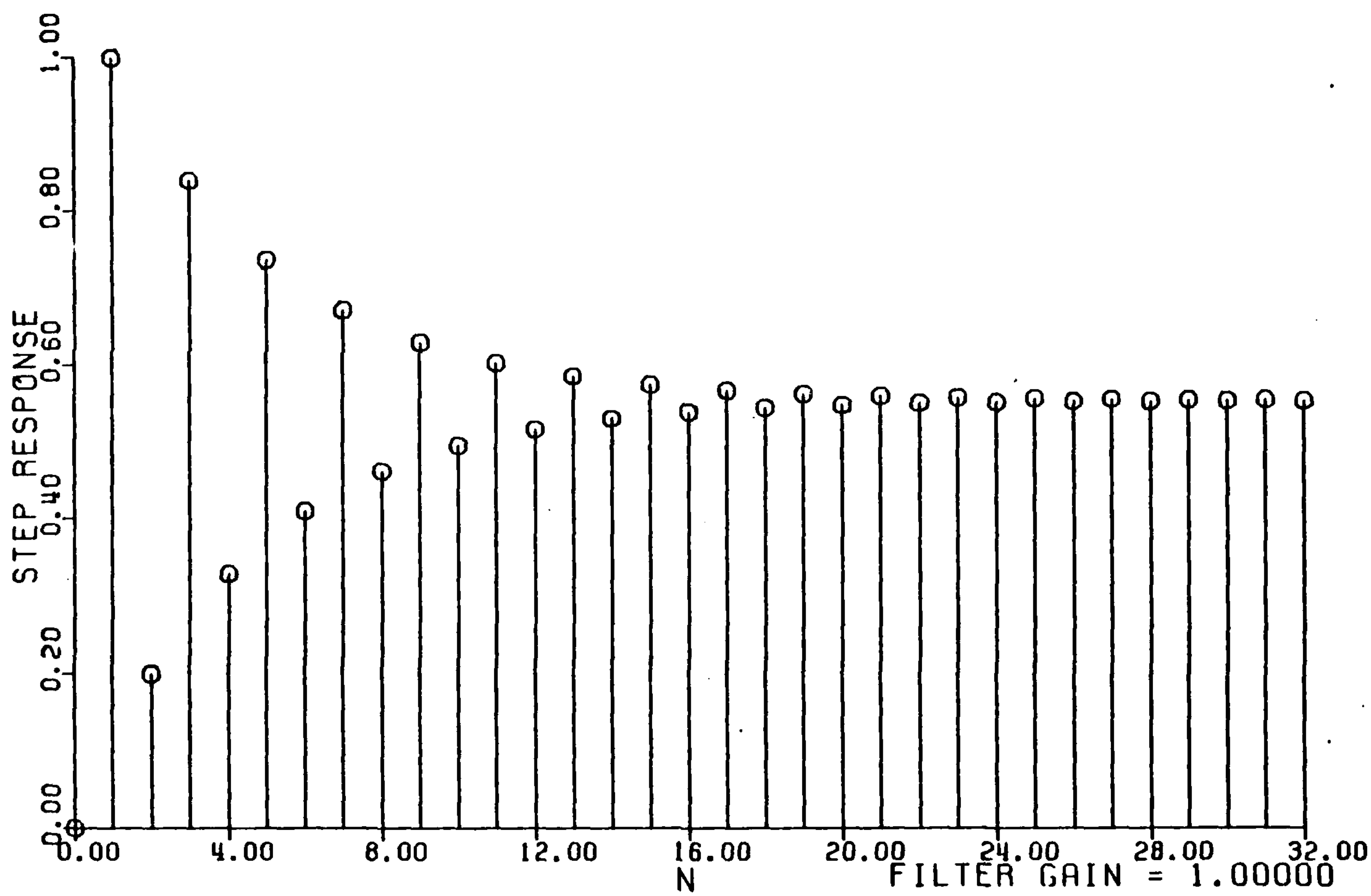


Figure 7-6c Unit Step Response For A Single Real Pole At (-0.8)

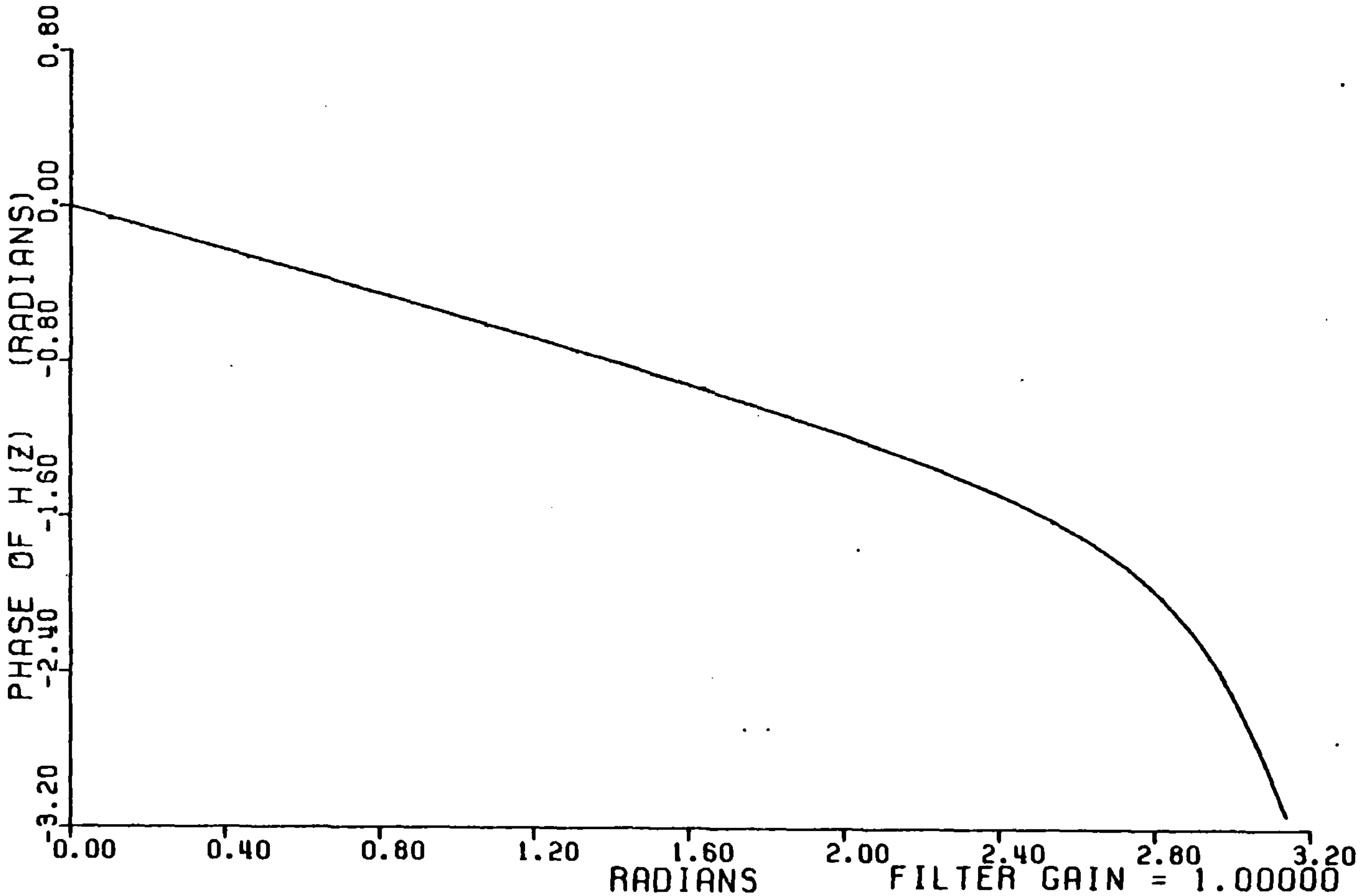


Figure 7-7d Phase Response For A Single Real Pole At (-0.8)

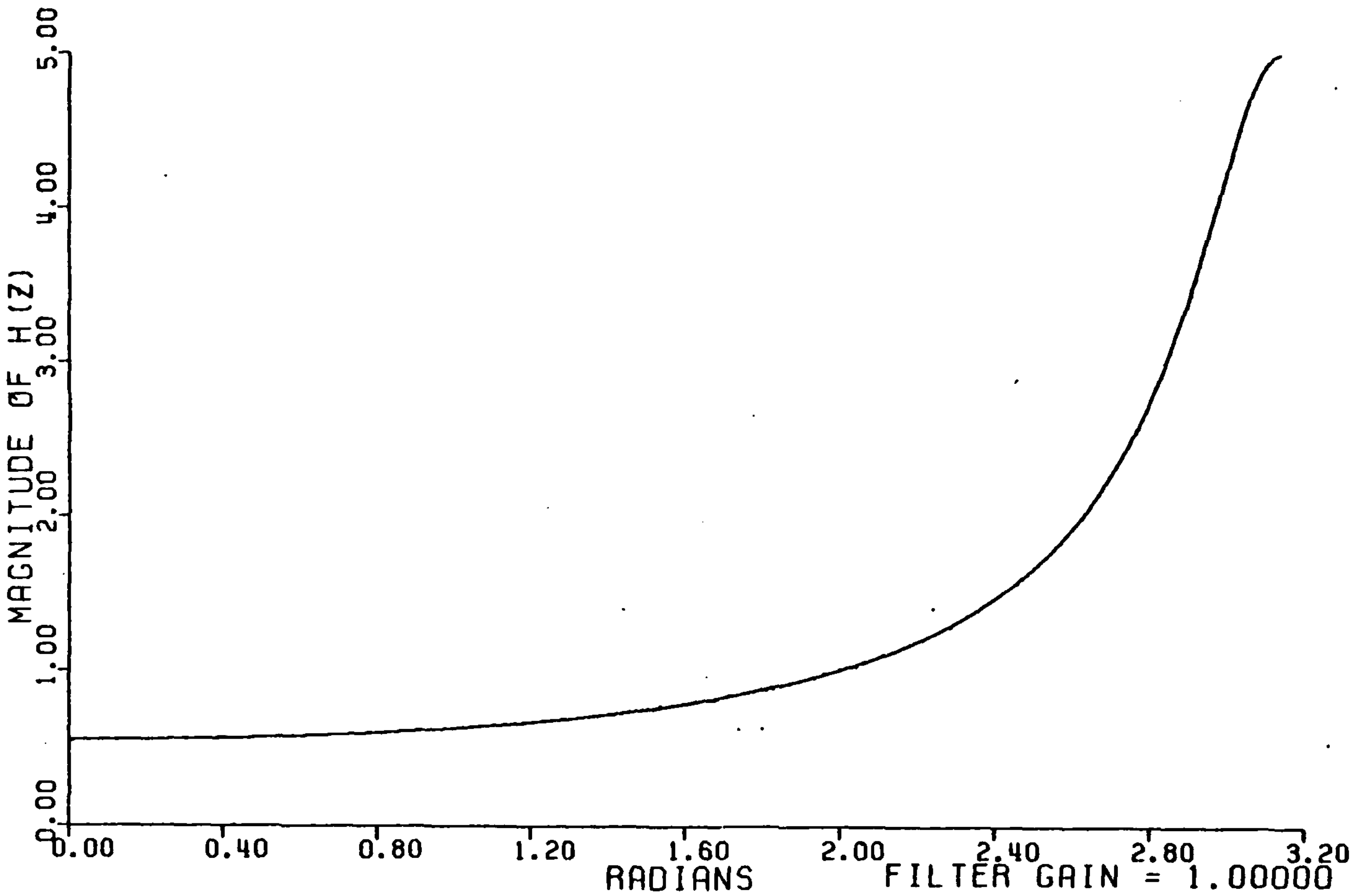


Figure 7-6e Magnitude Of H(z) For A Single Real Pole At (-0.8)

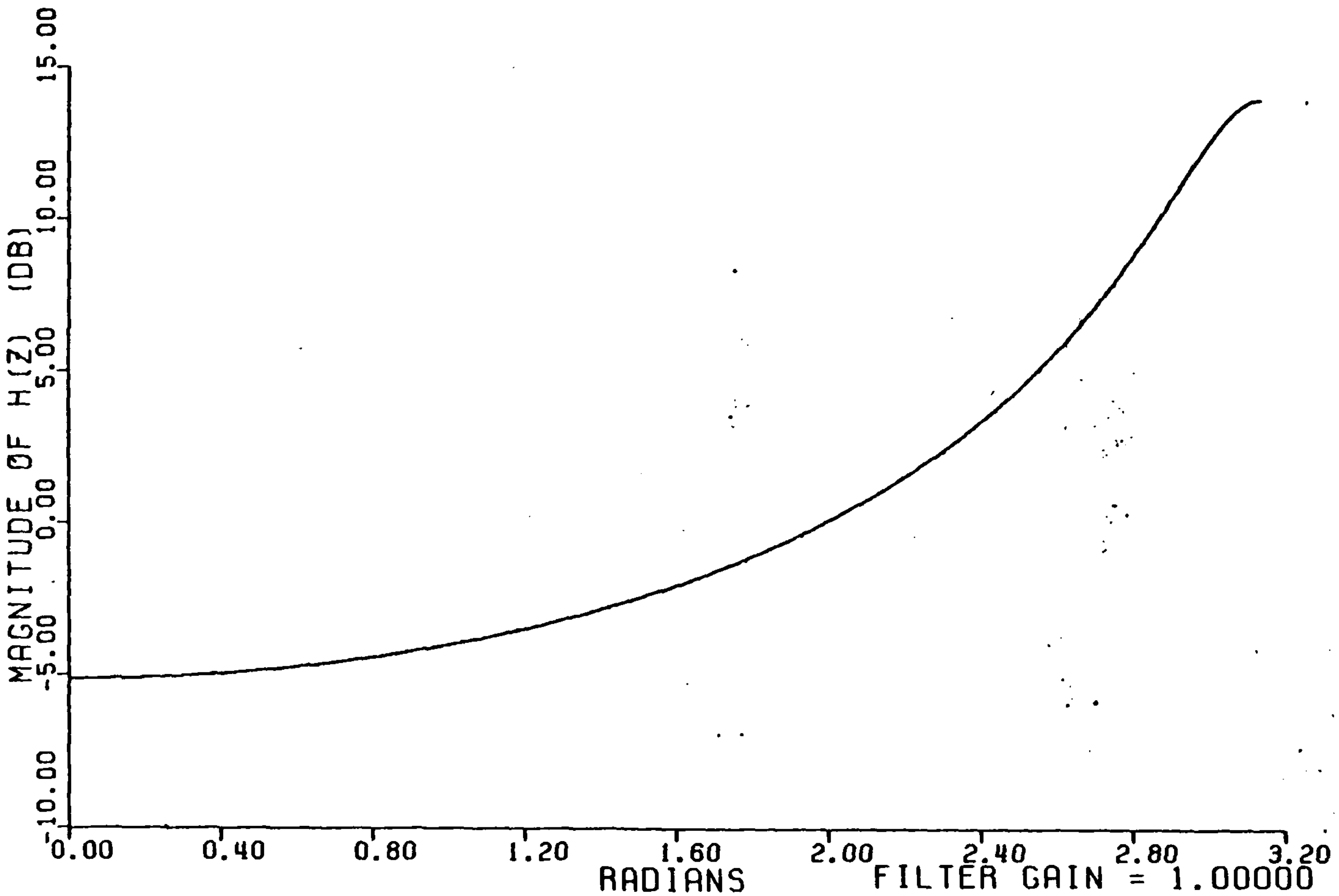


Figure 7-6f Magnitude Of *H(z)* In *Decibels For a Single* Real Pole at (-0.8)

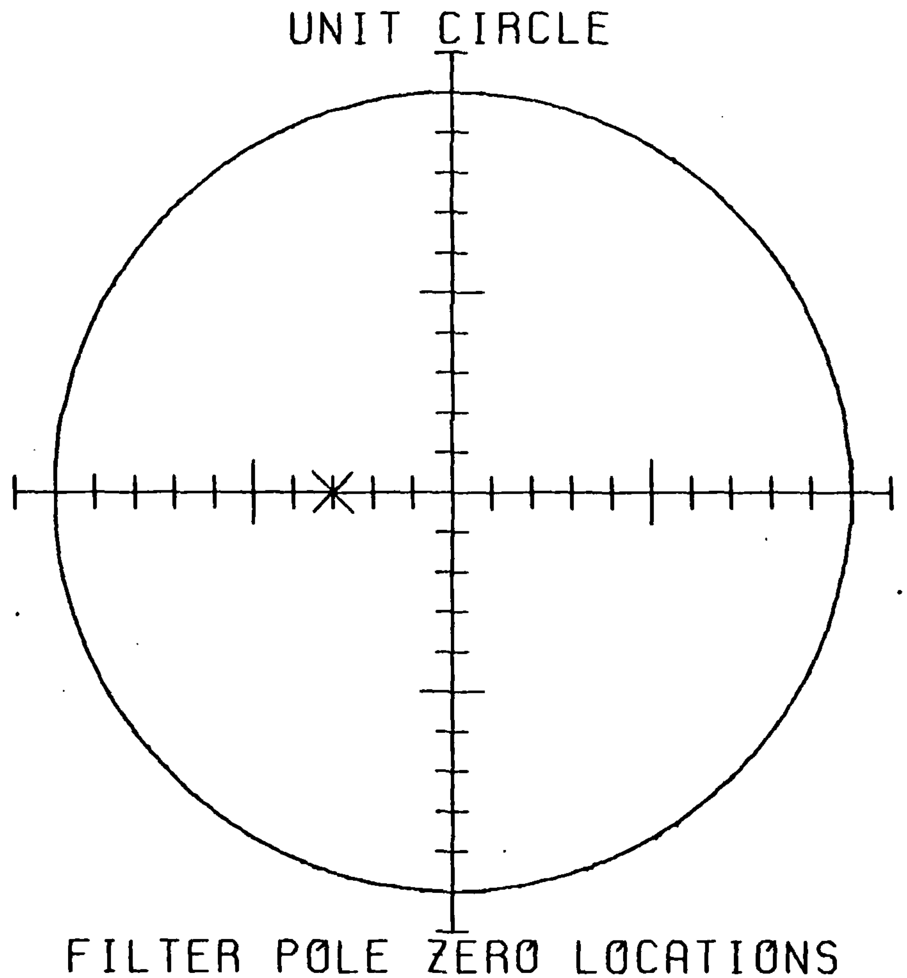


Figure 7-7a Example Of A Single Real Pole At (-0.3)

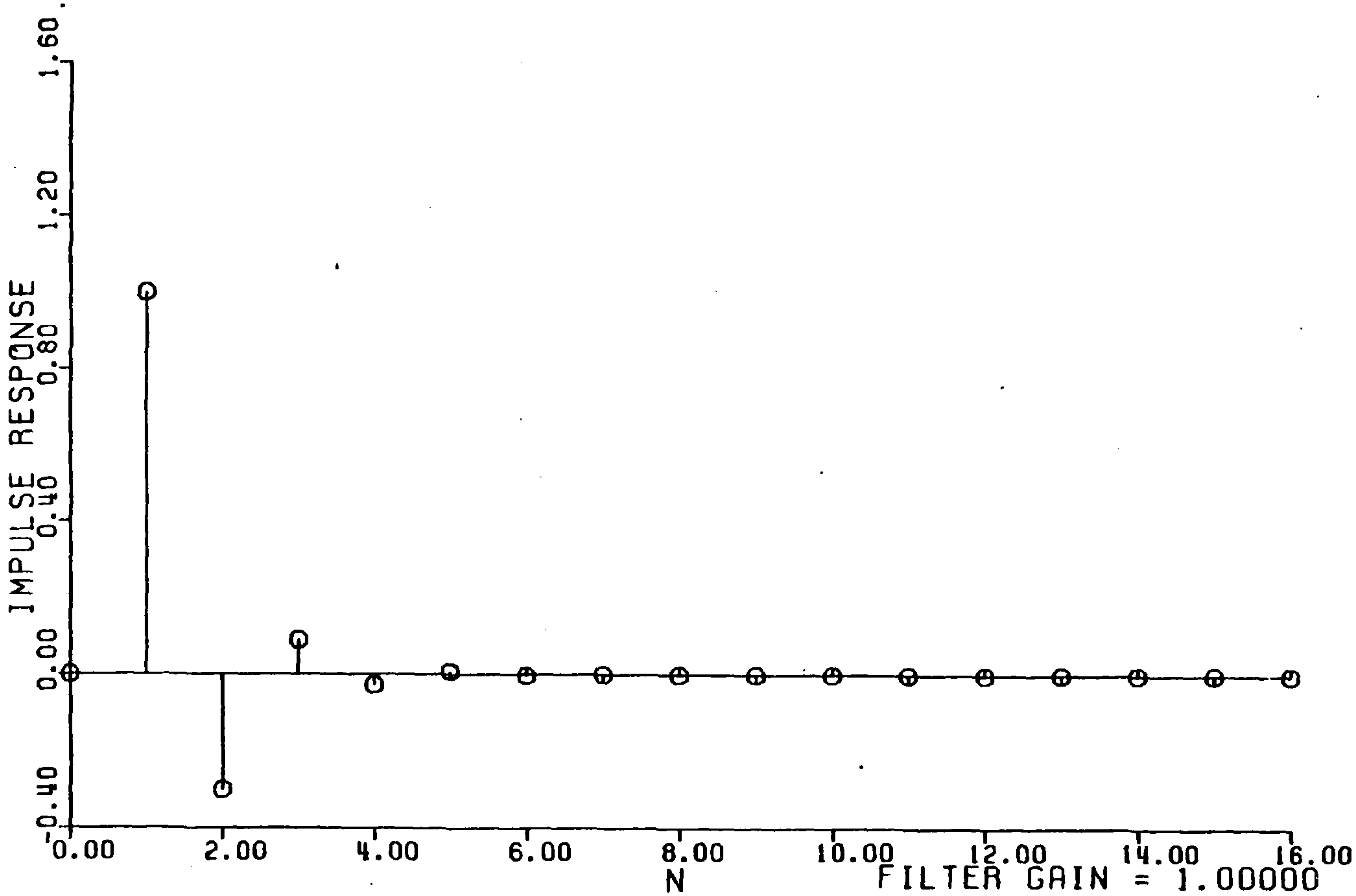


Figure 7-7b Unit Sample Response For A Single Real Pole At (-0.3)

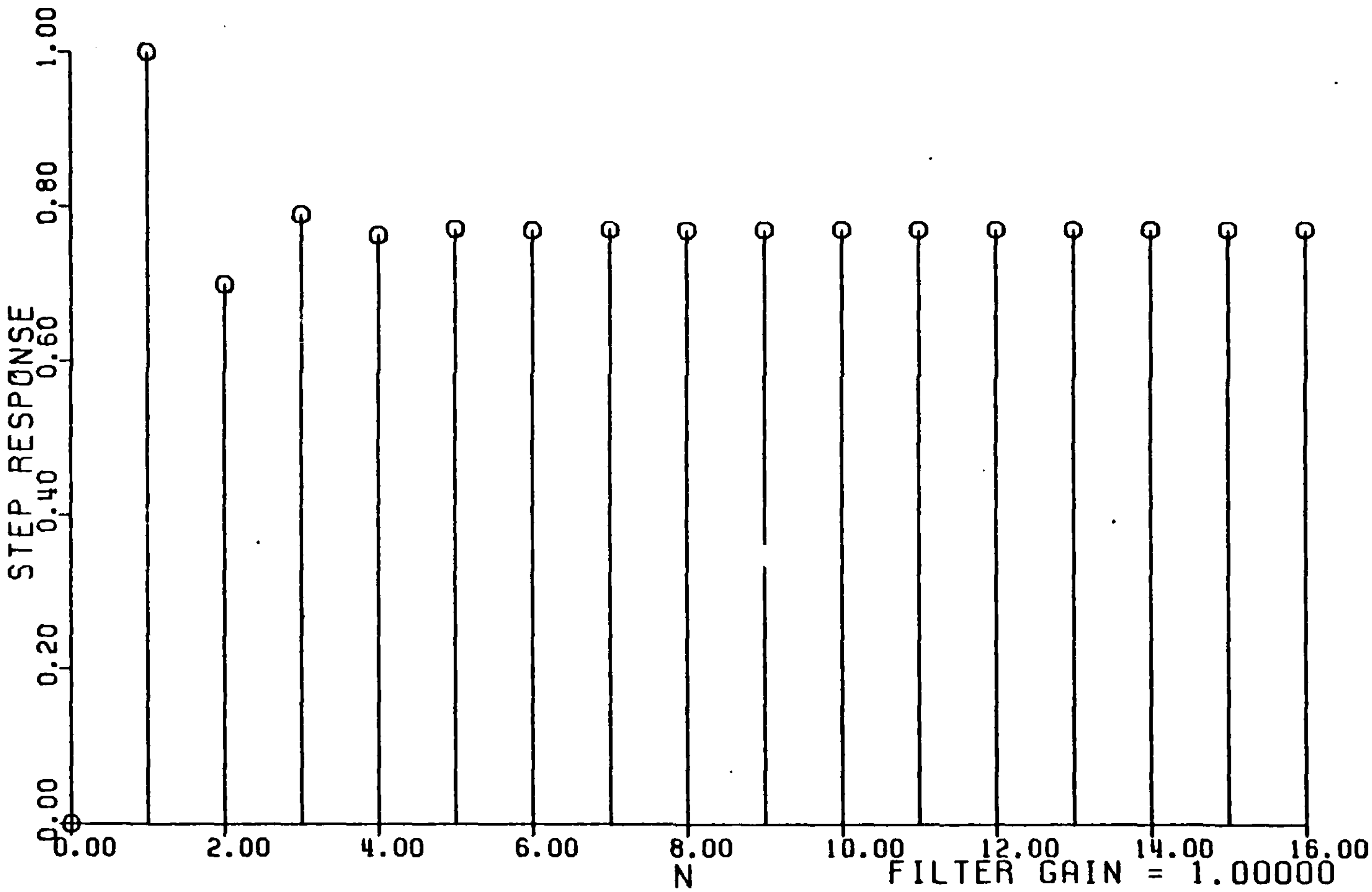


Figure 7-7c Unit Step Response For A Single Real Pole At (-0.3)

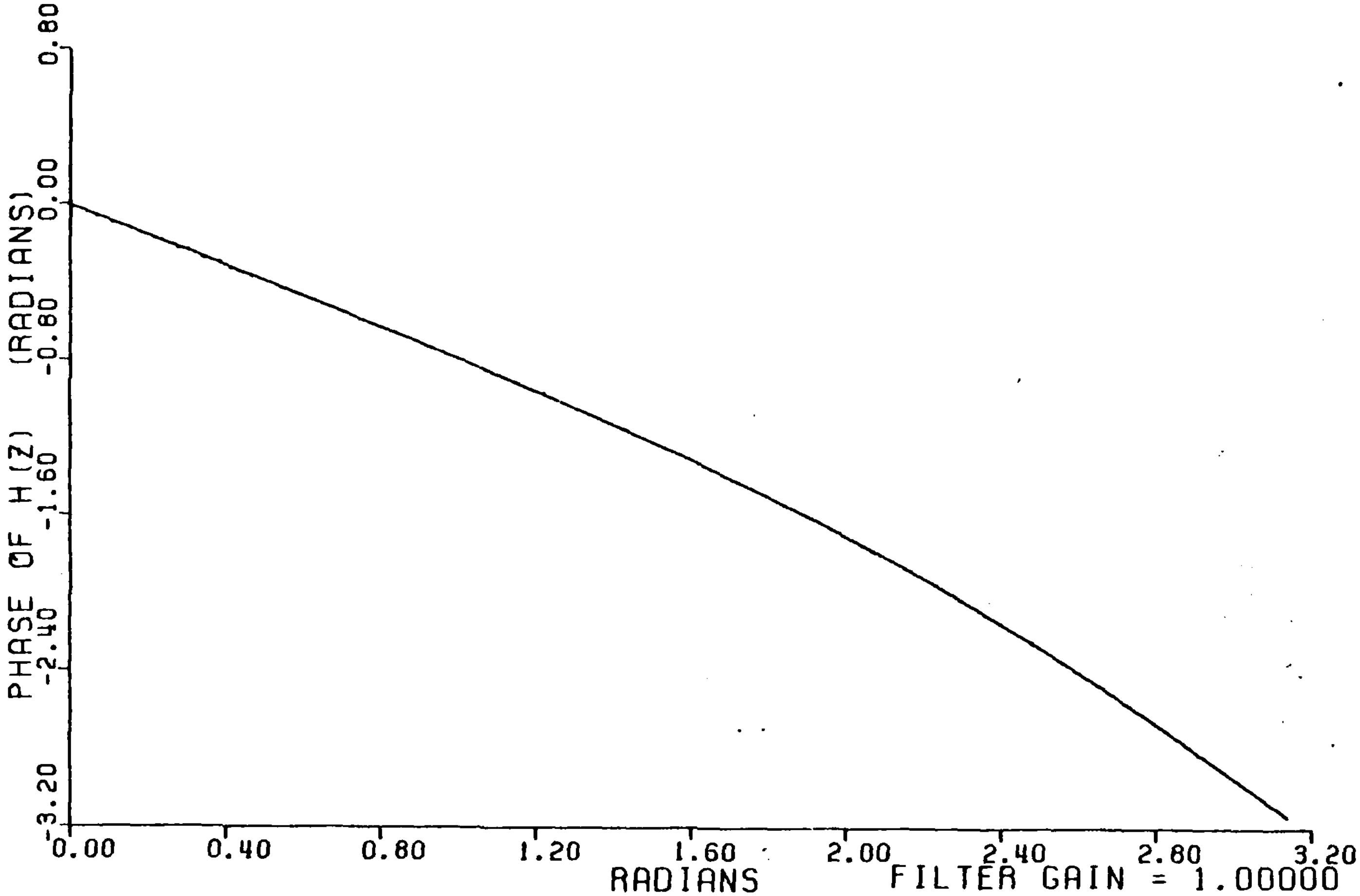


Figure 7-7d Phase Response For A Single Real Pole At (-0.3)

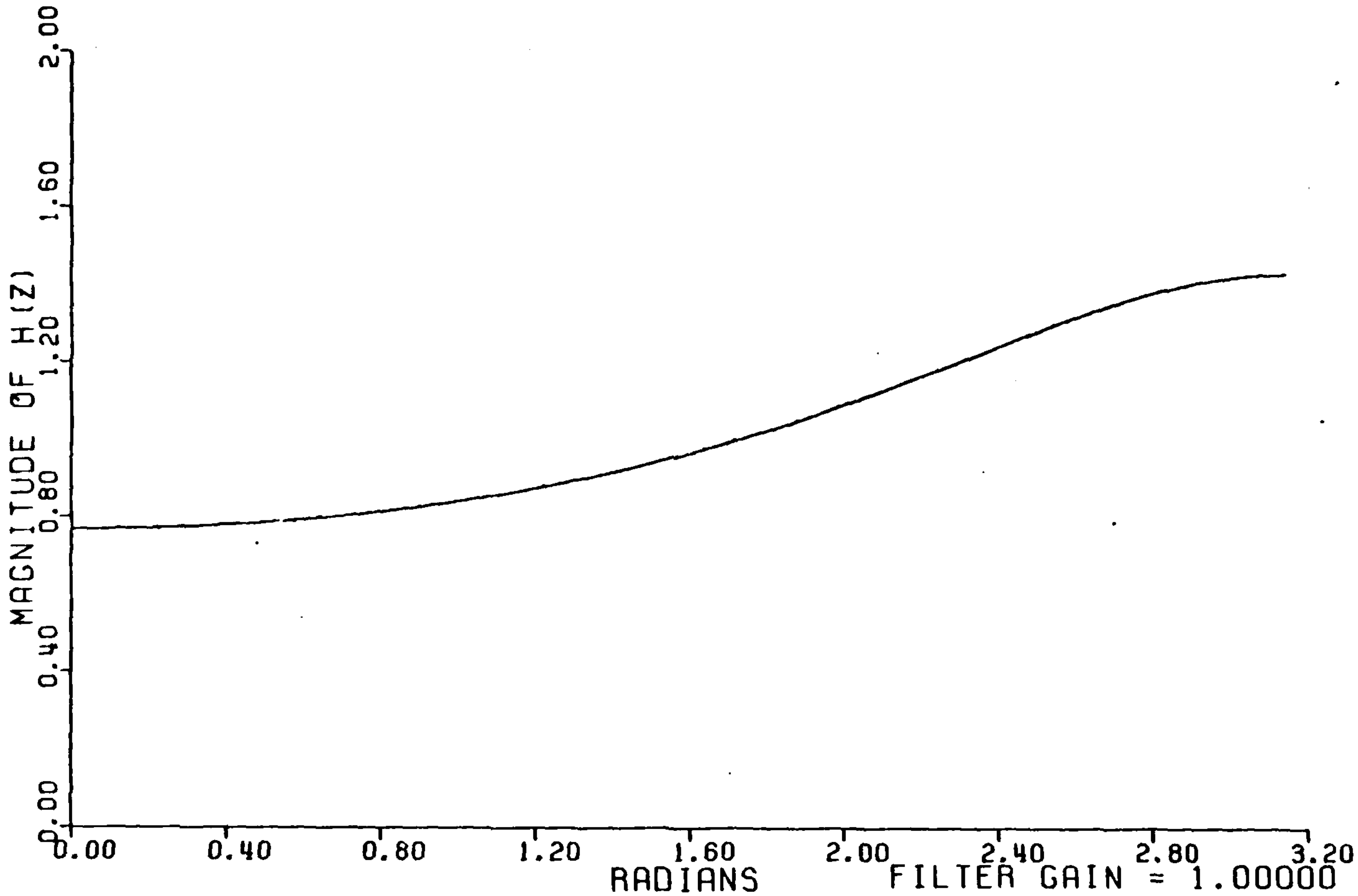


Figure 7-7e Magnitude Of H(z) For A Single Real Pole At (-0.3)

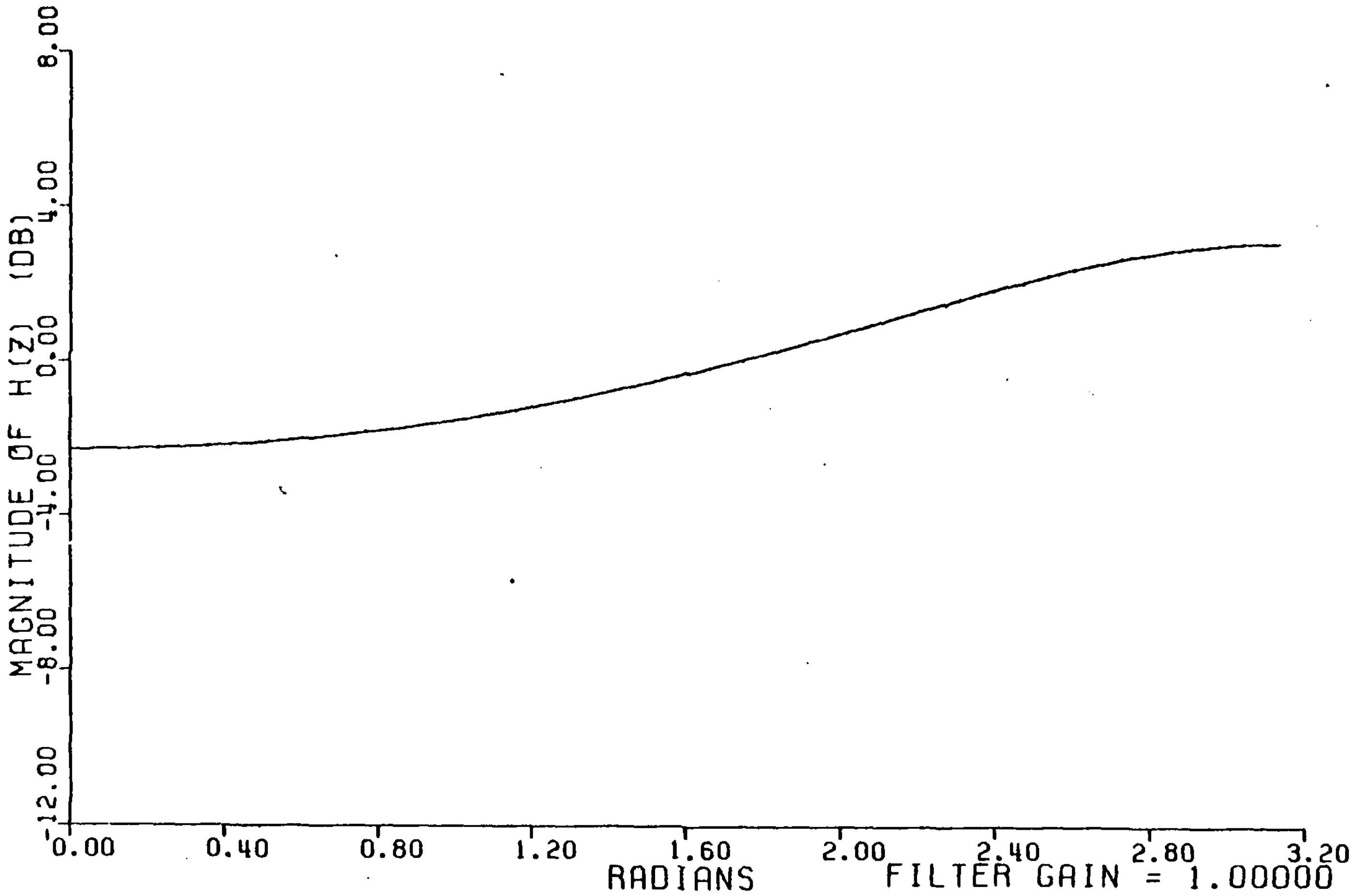


Figure 7-7f Magnitude Of H(z) For A Single Real Pole At (-0.3)

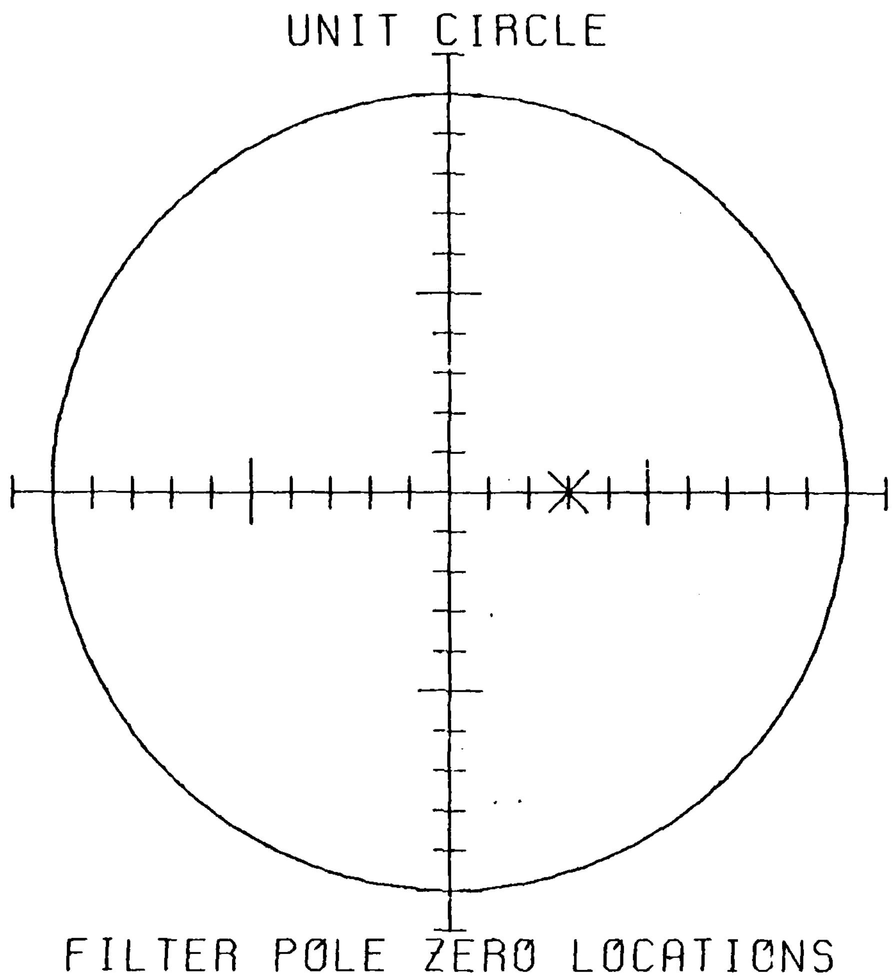


Figure 7-8a Example Of A Single Real Pole At (0.3)

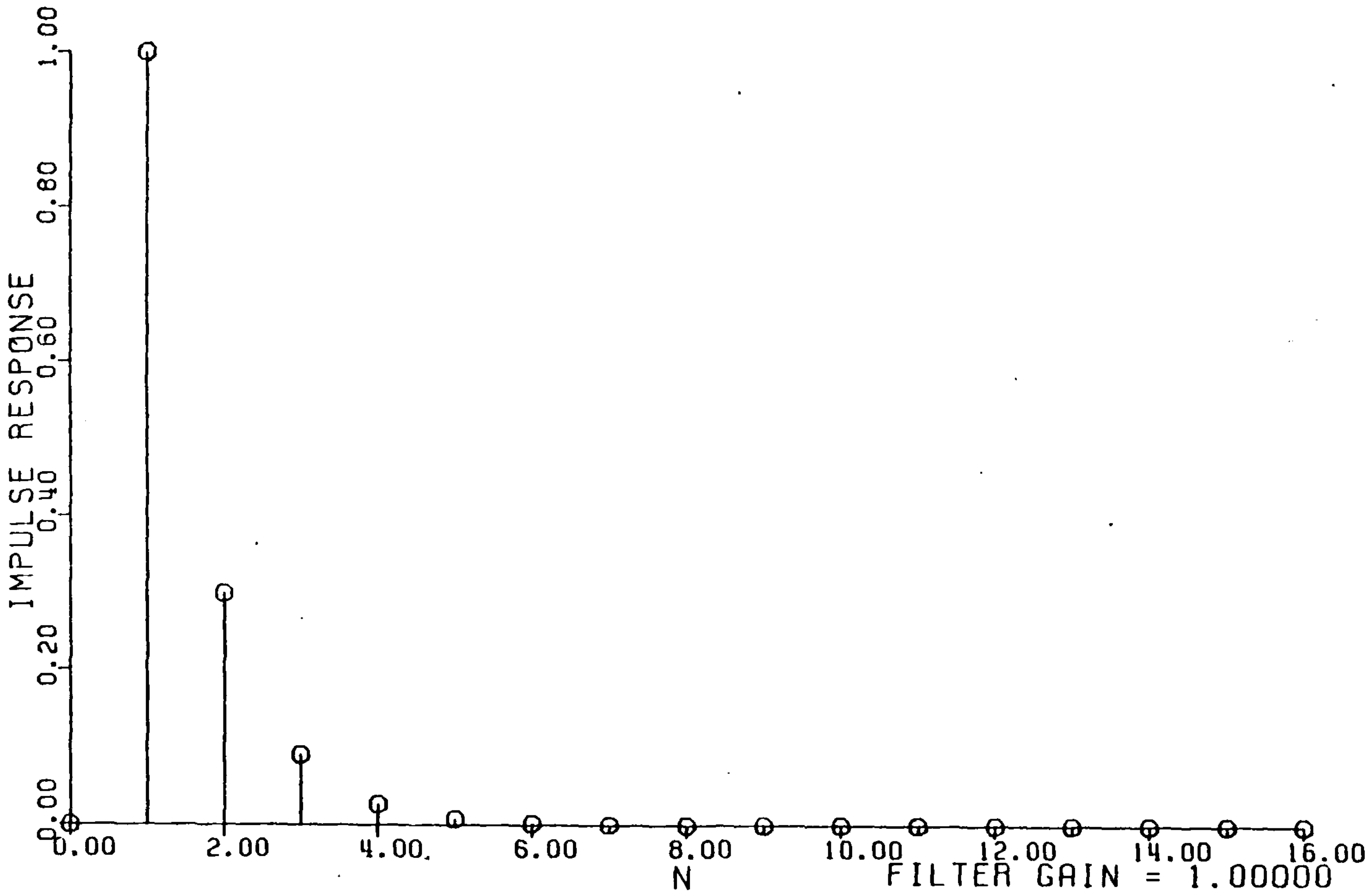


Figure 7-8b Unit Sample Response For A Single Real Pole At (0.3)

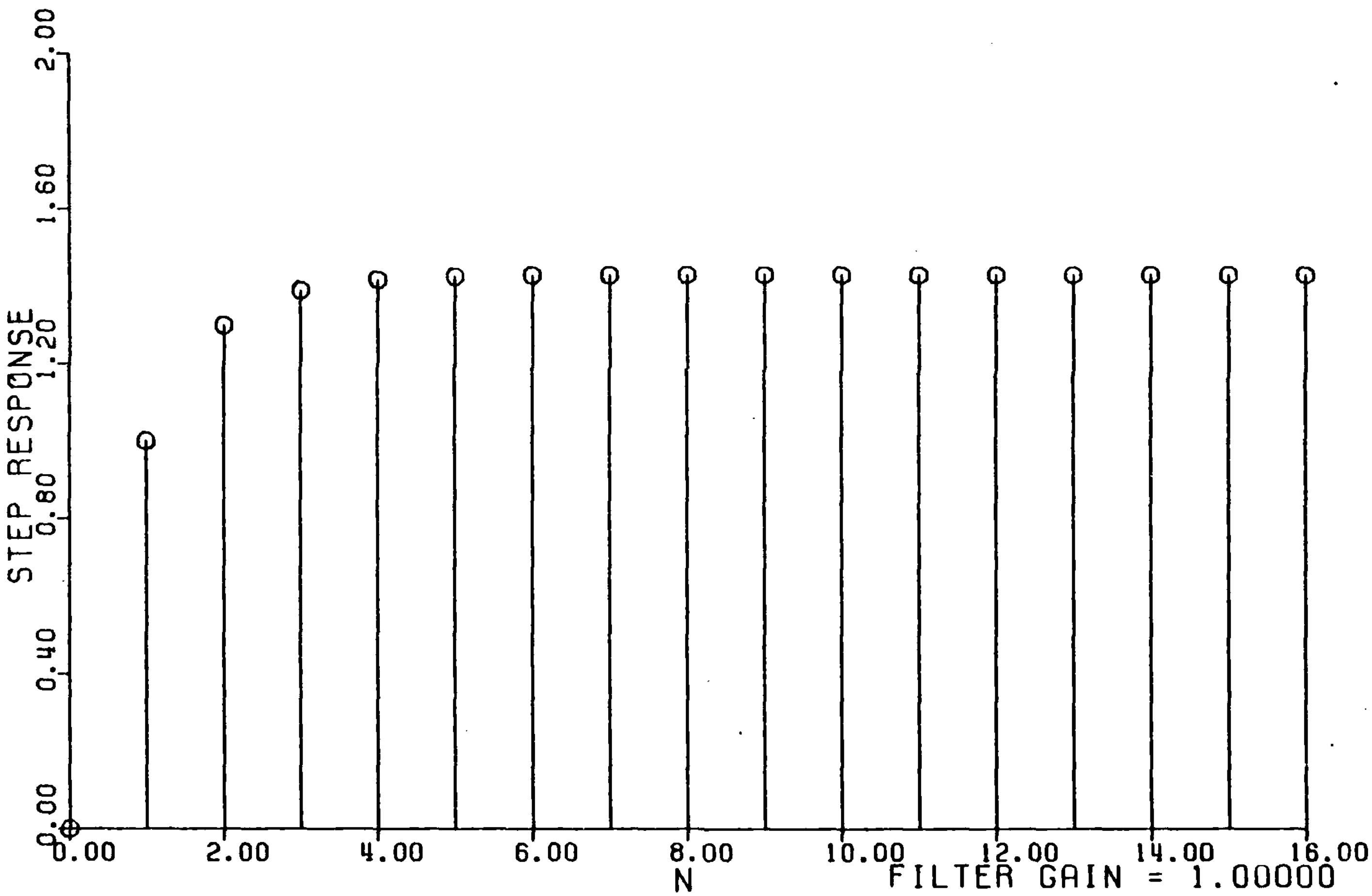


Figure 7-8c Unit Step Response For A Single Real Pole At (0.3)

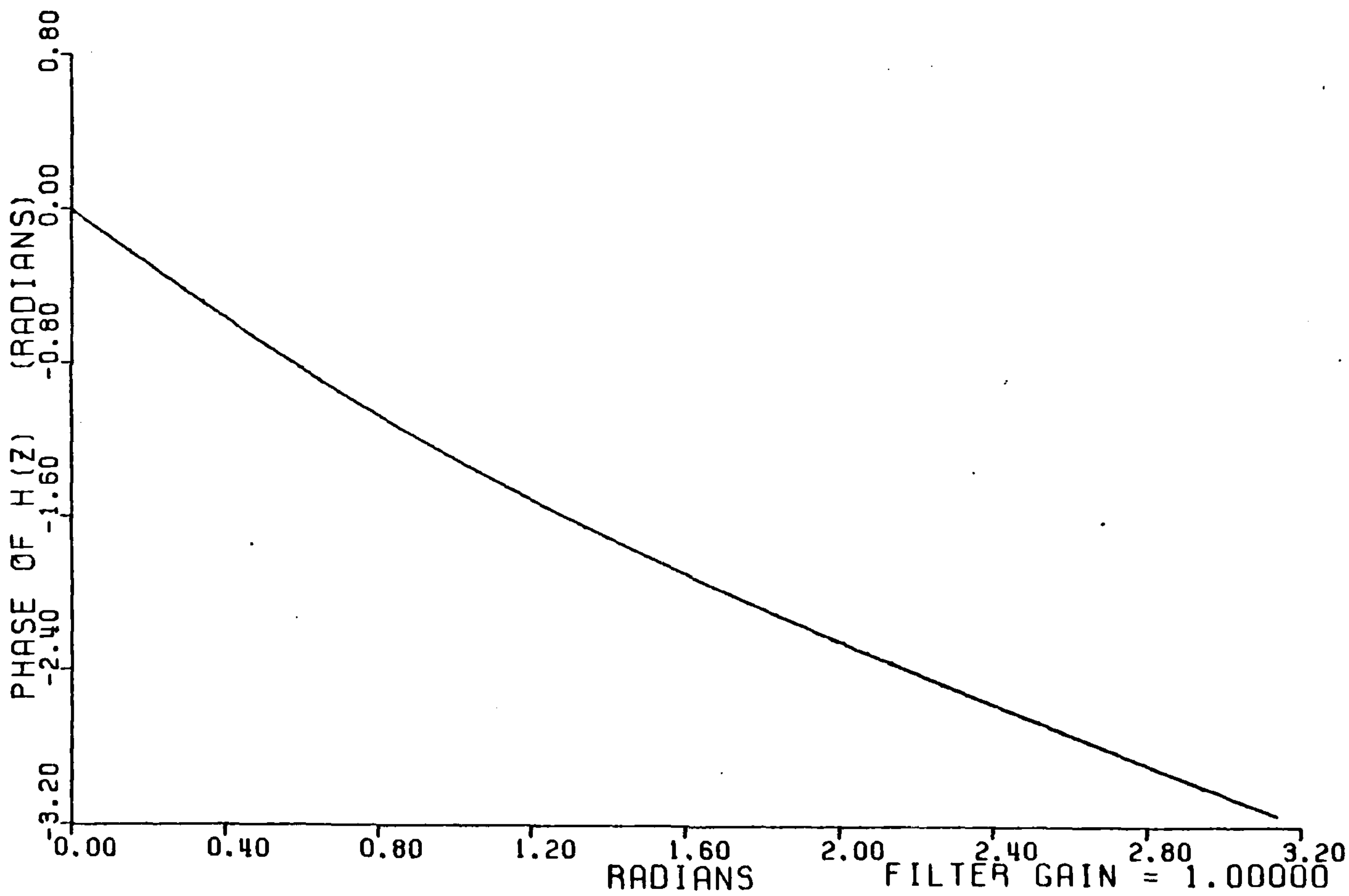


Figure 7-8d Phase Response For A Single Real Pole At (0.3)

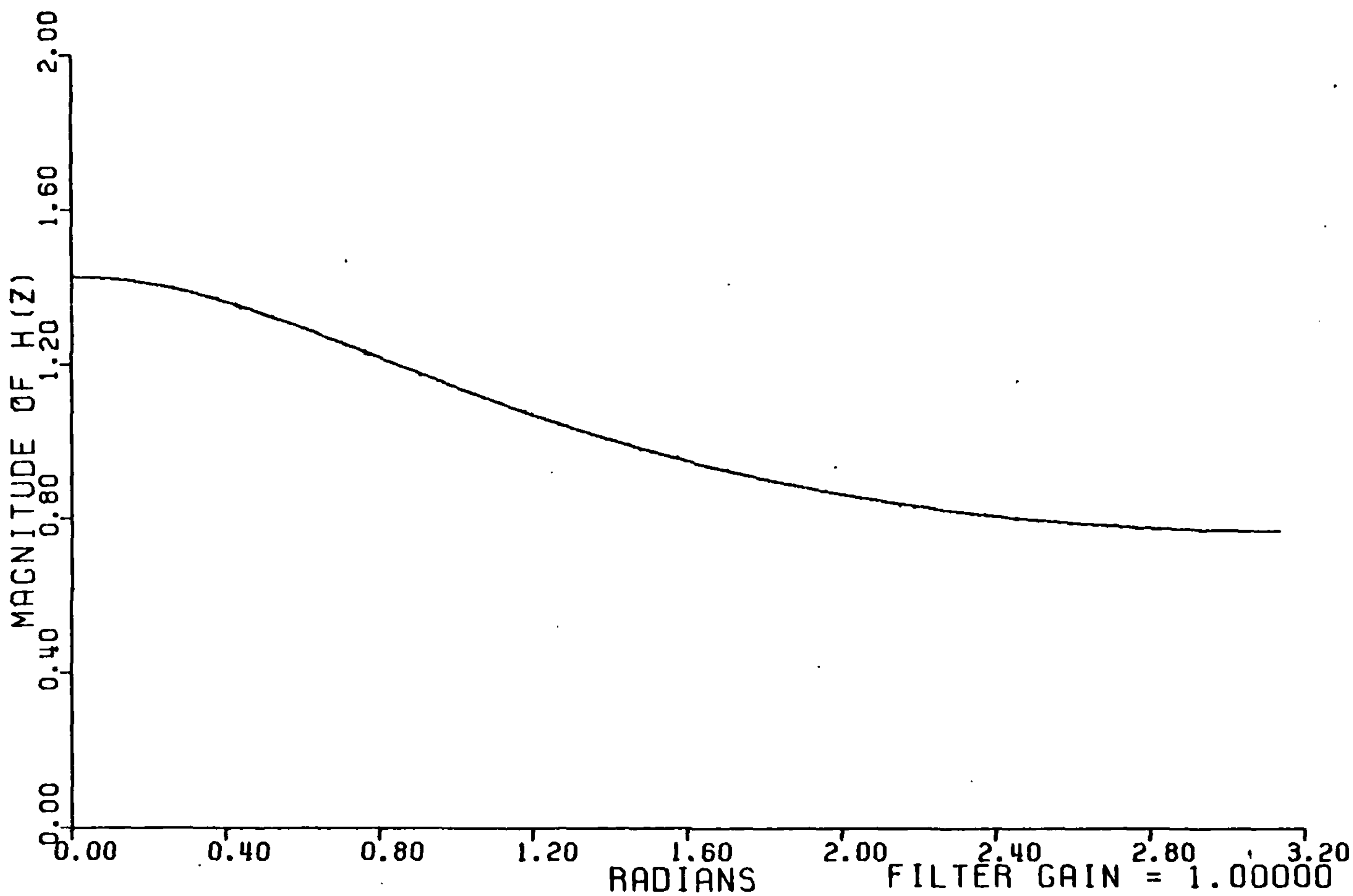


Figure 7-8e Magnitude of H(z) For A Single Real Pole At (0.3)

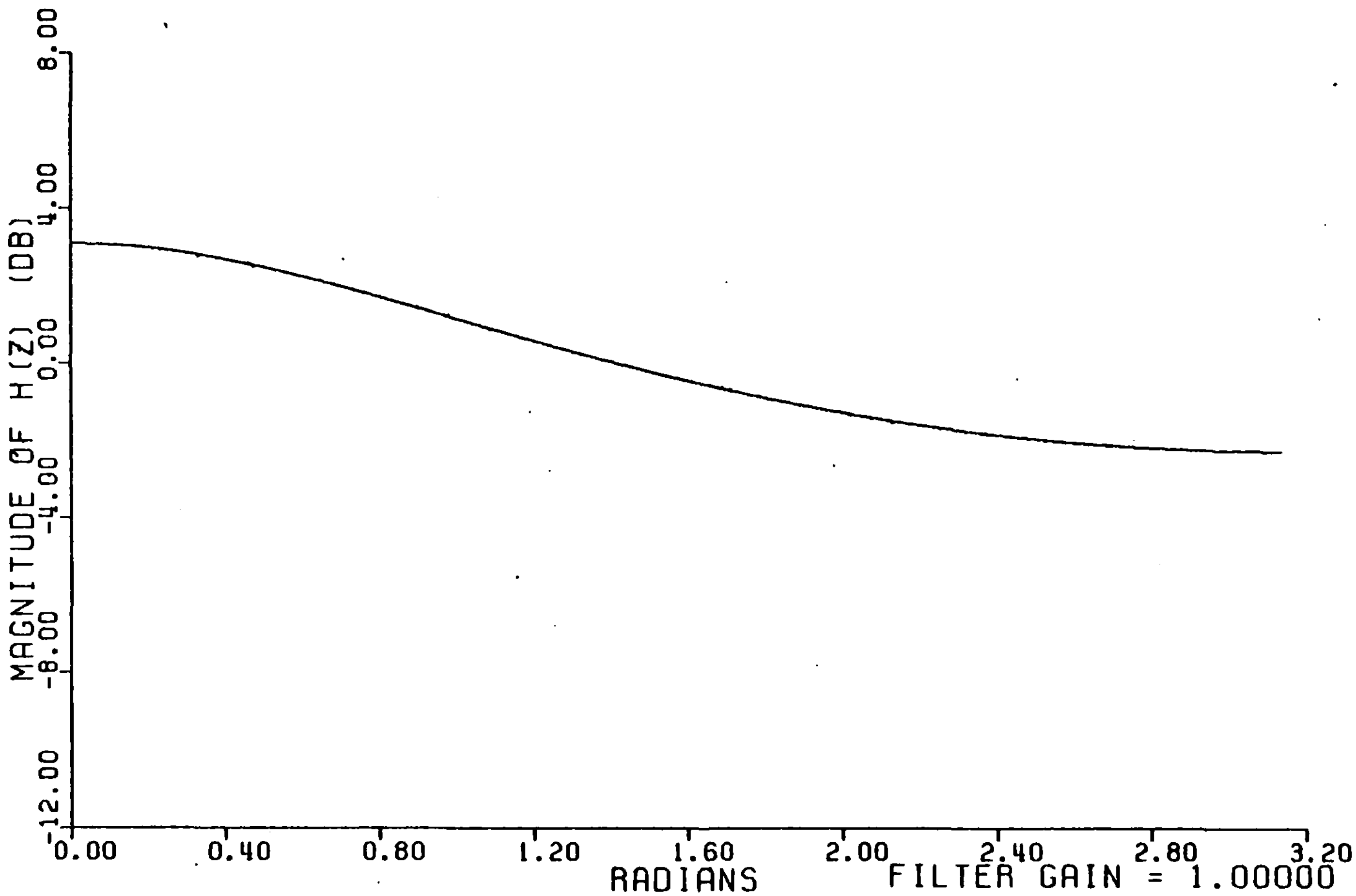


Figure 7-8f Magnitude Of H(z) In Decibels For A Single Real Pole At (0.3)

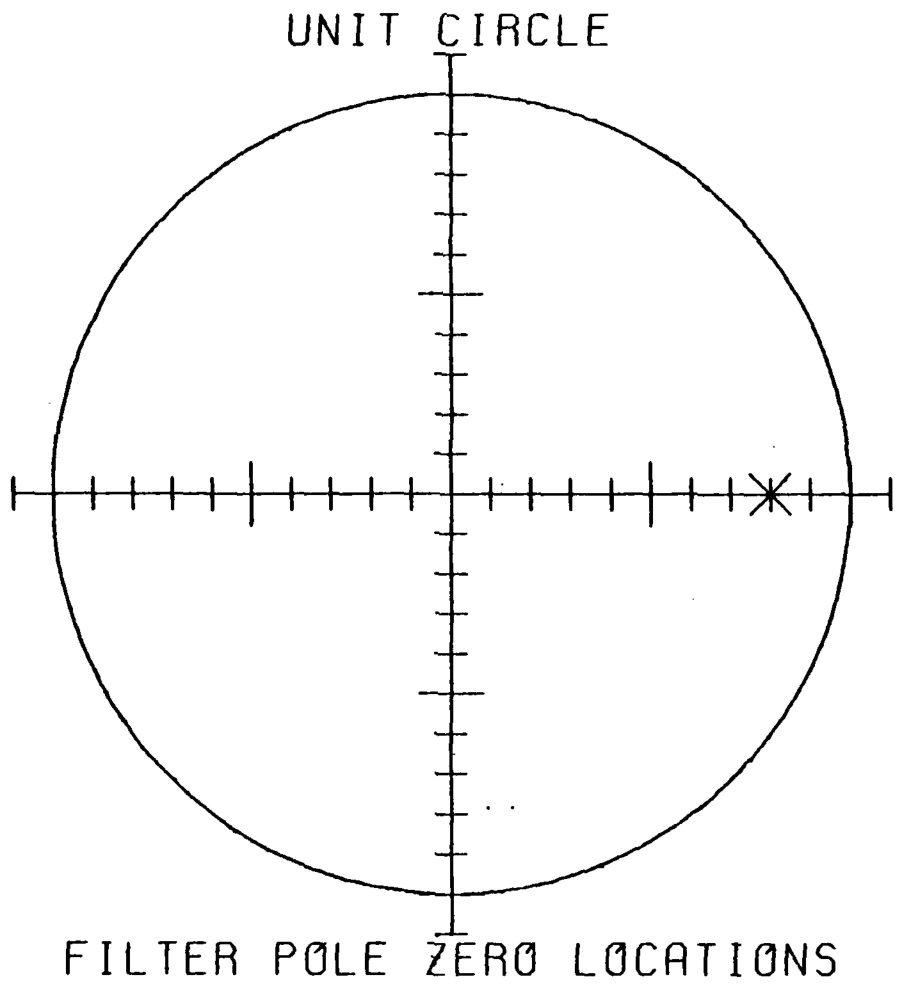


Figure 7-9a Example Of A Single Real Pole At (0.8)

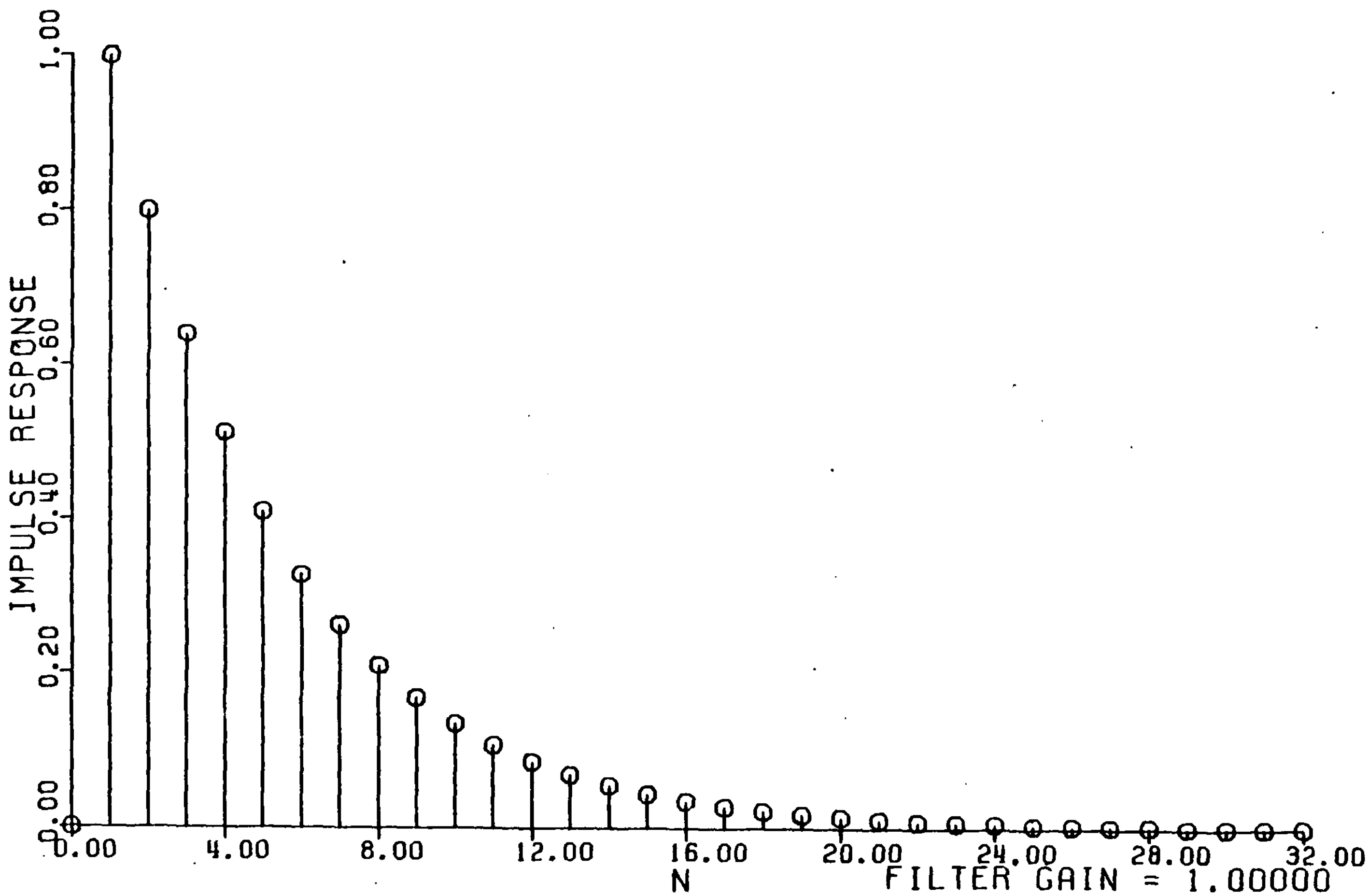


Figure 7-9b Unit Sample Response For A Single Real Pole At (0.8)

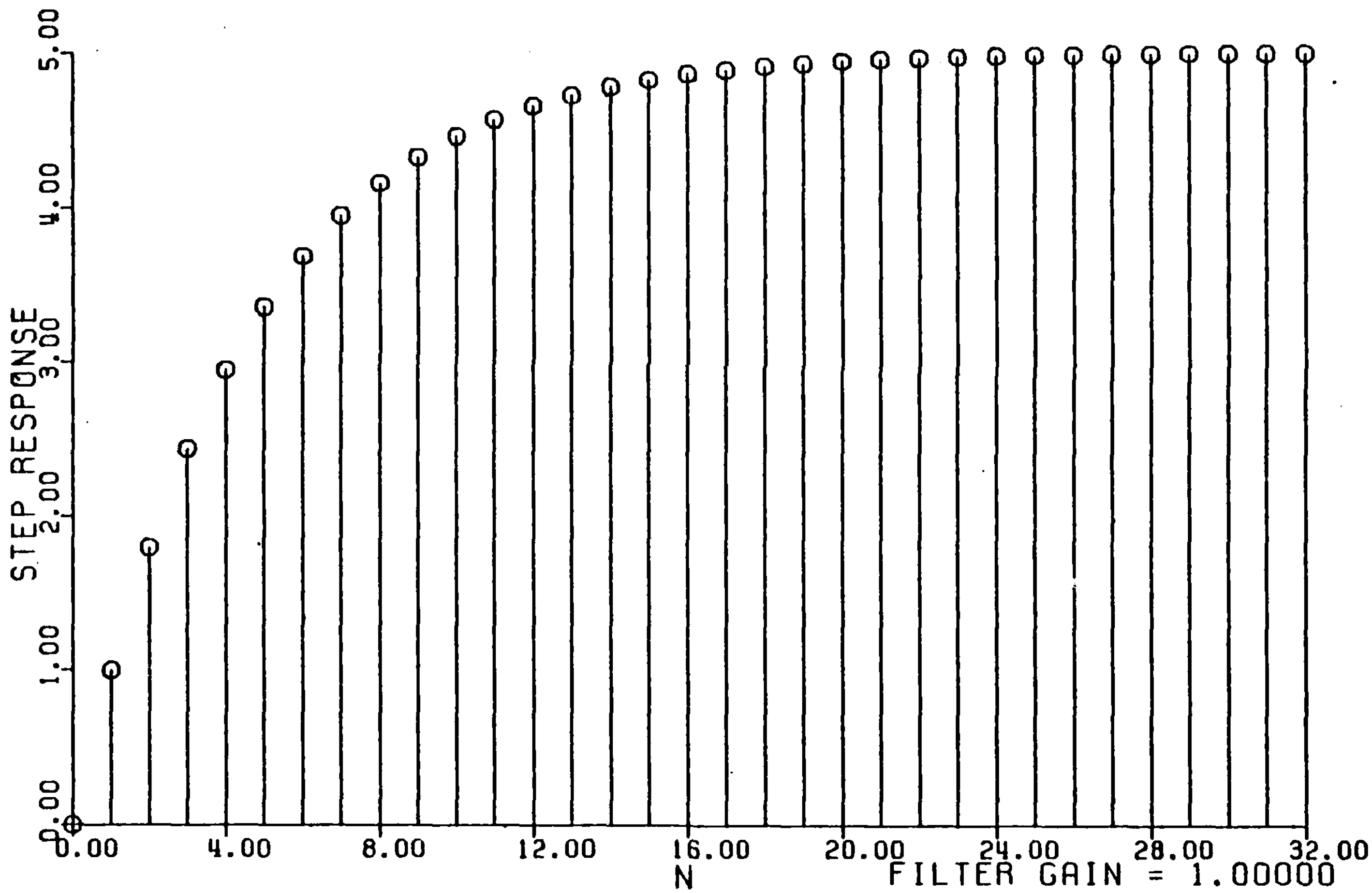


Figure 7-9c Unit Step Response For A Single Real Pole At (0.8)

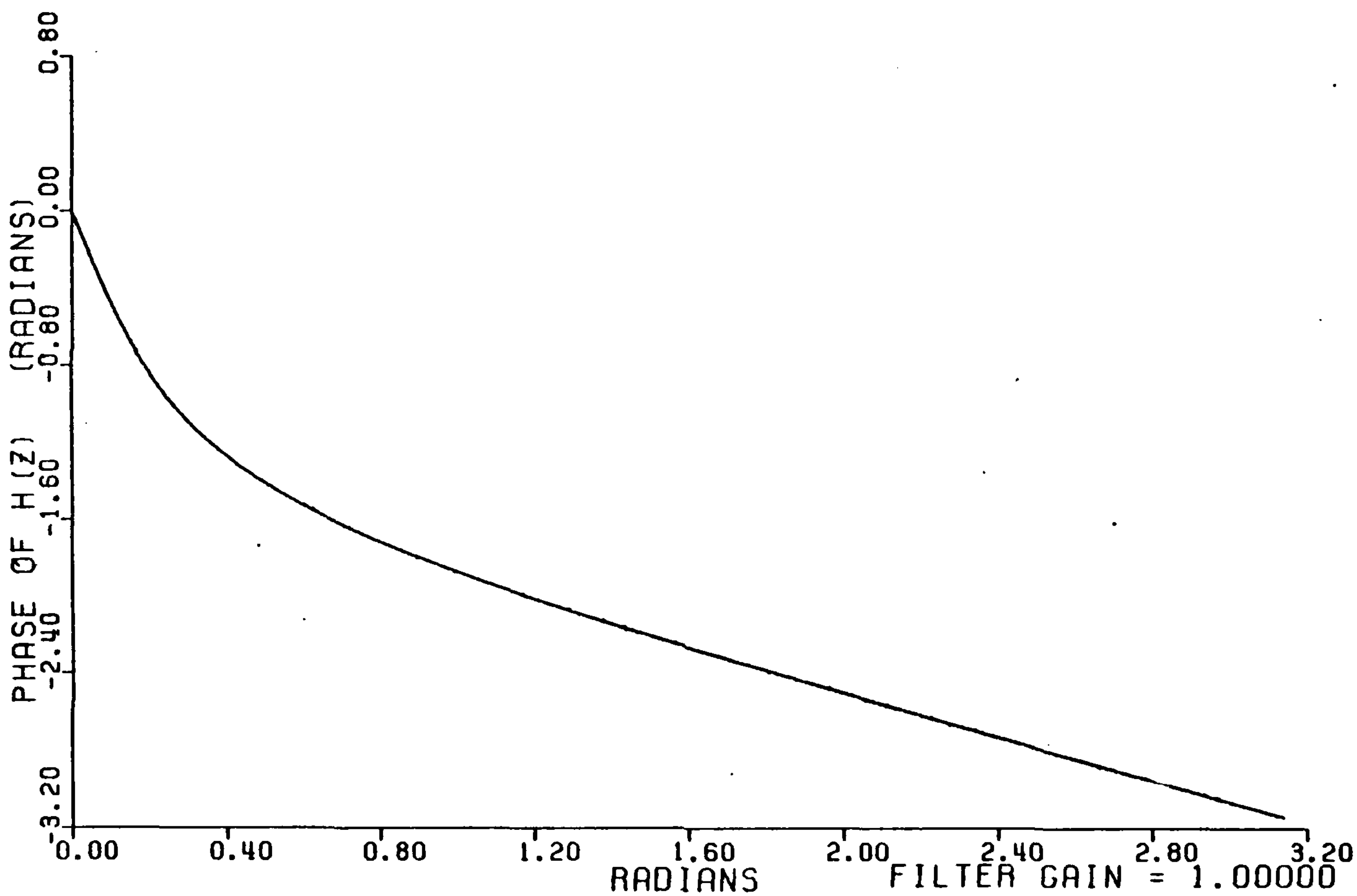


Figure 7-9d Phase Response For A Single Real Pole At (0.8)

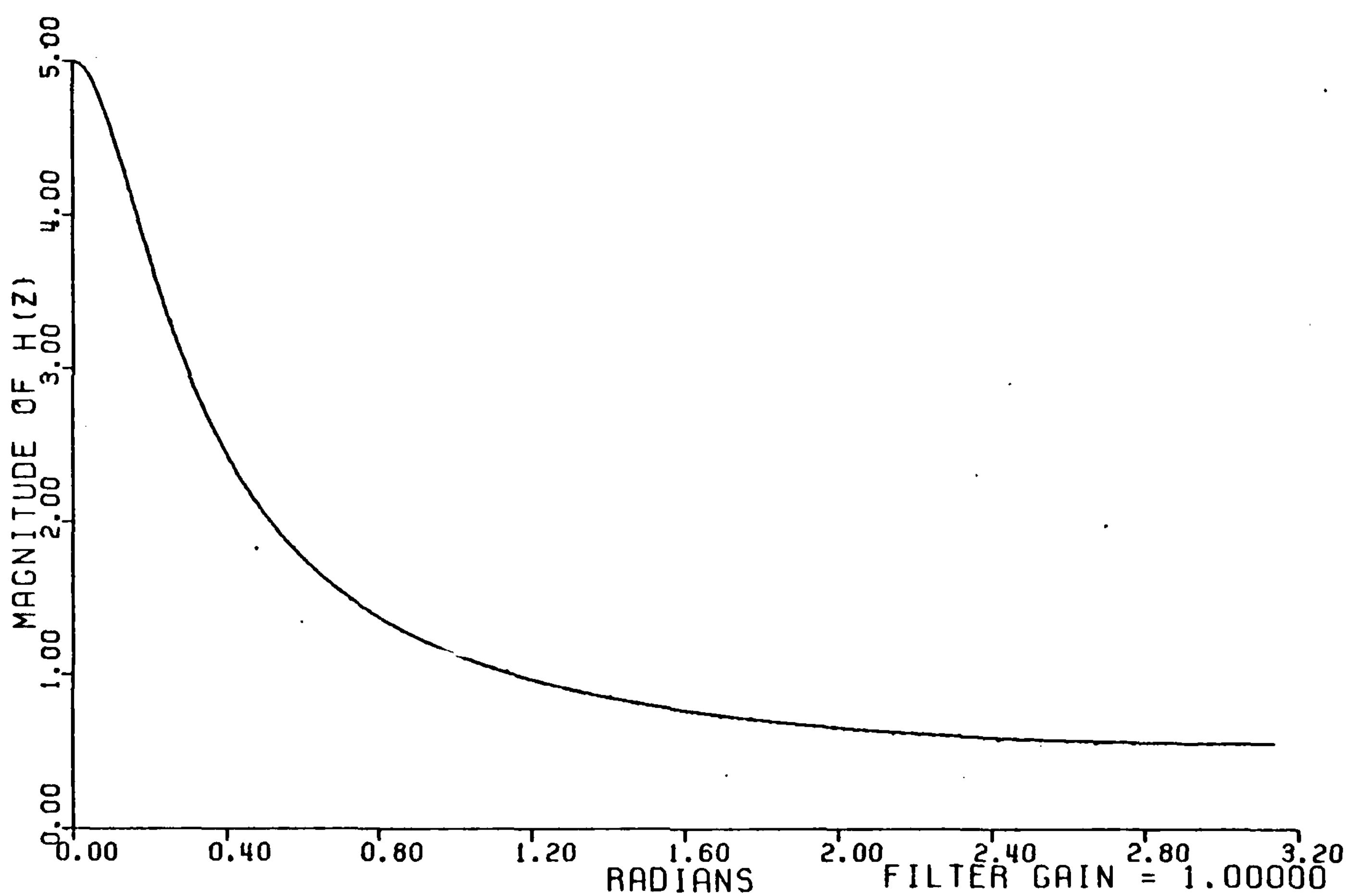


Figure 7-9e Magnitude Of H(z) For A Single Real Pole At (0.8)

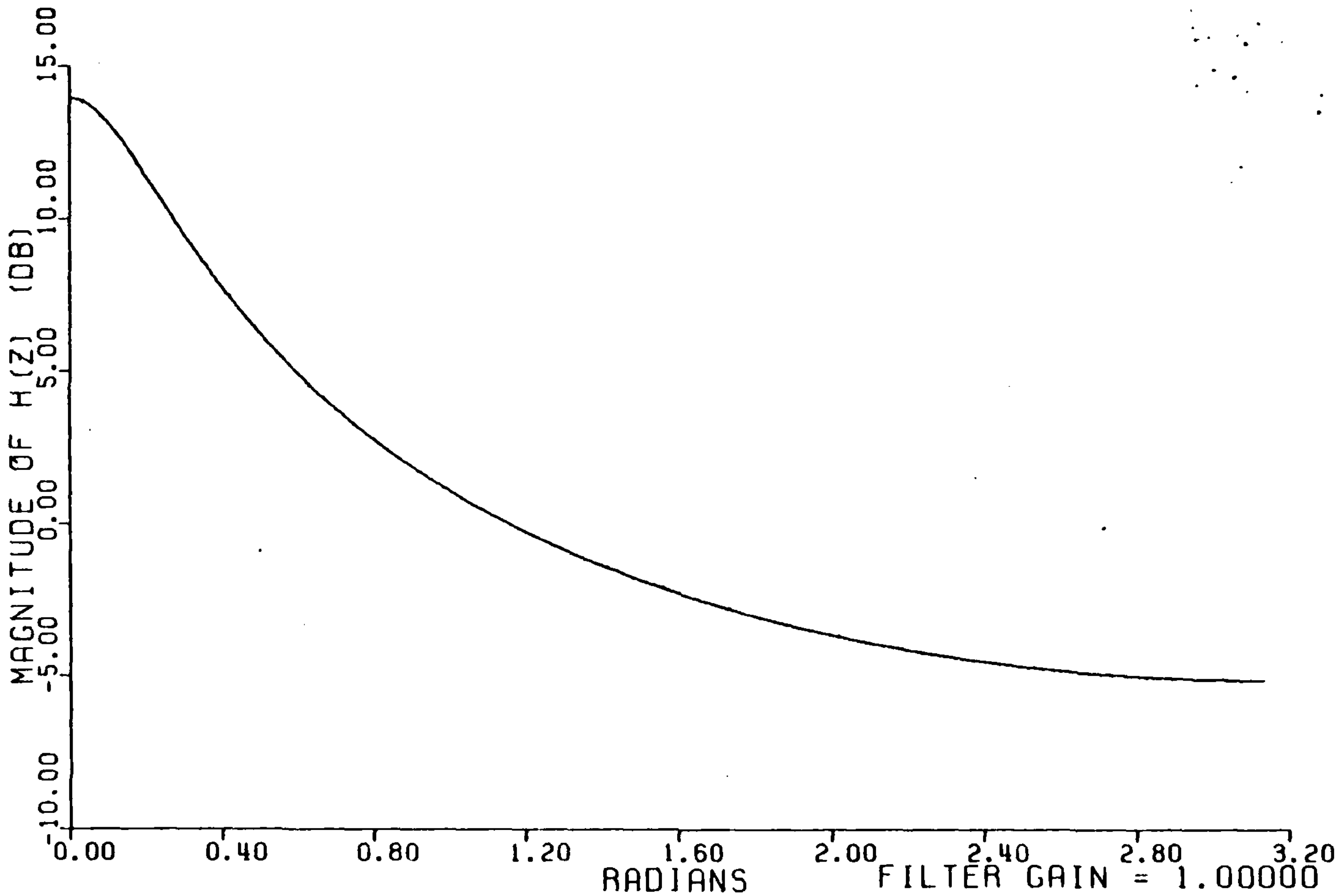
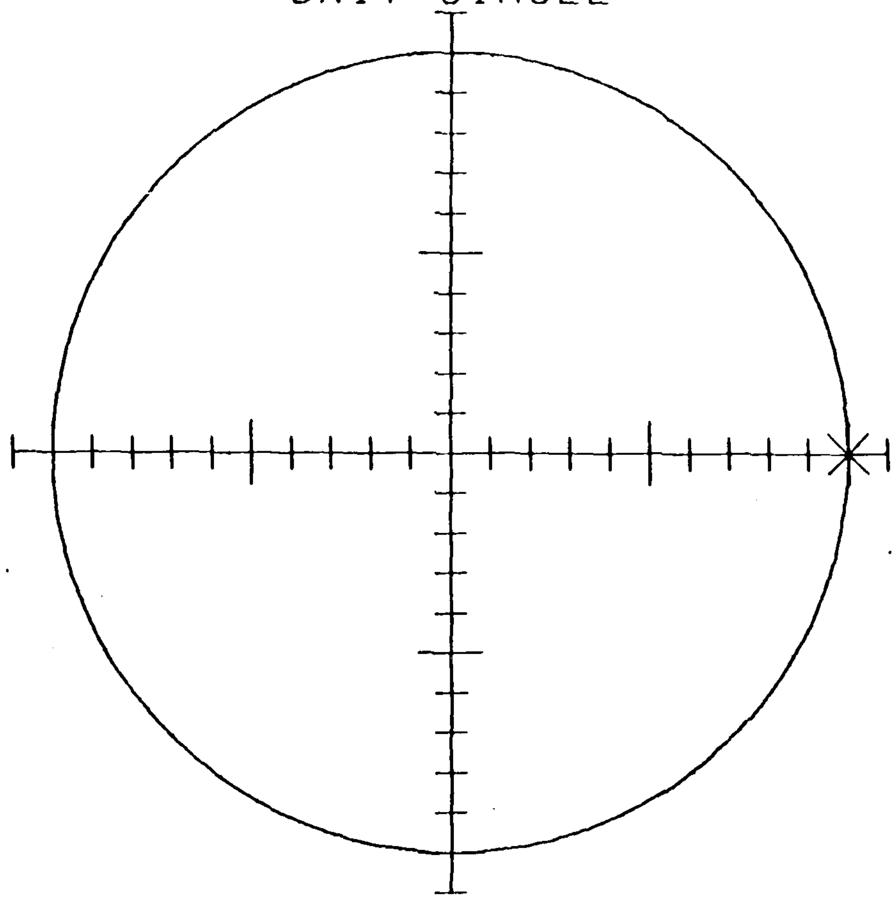


Figure 7-9f Magnitude Of H(z) For A Single Real Pole At (0.8)

UNIT CIRCLE



FILTER POLE ZERO LOCATIONS

Figure 7-10a Example Of A Single Real Pole On the Unit Circle at (1.0)

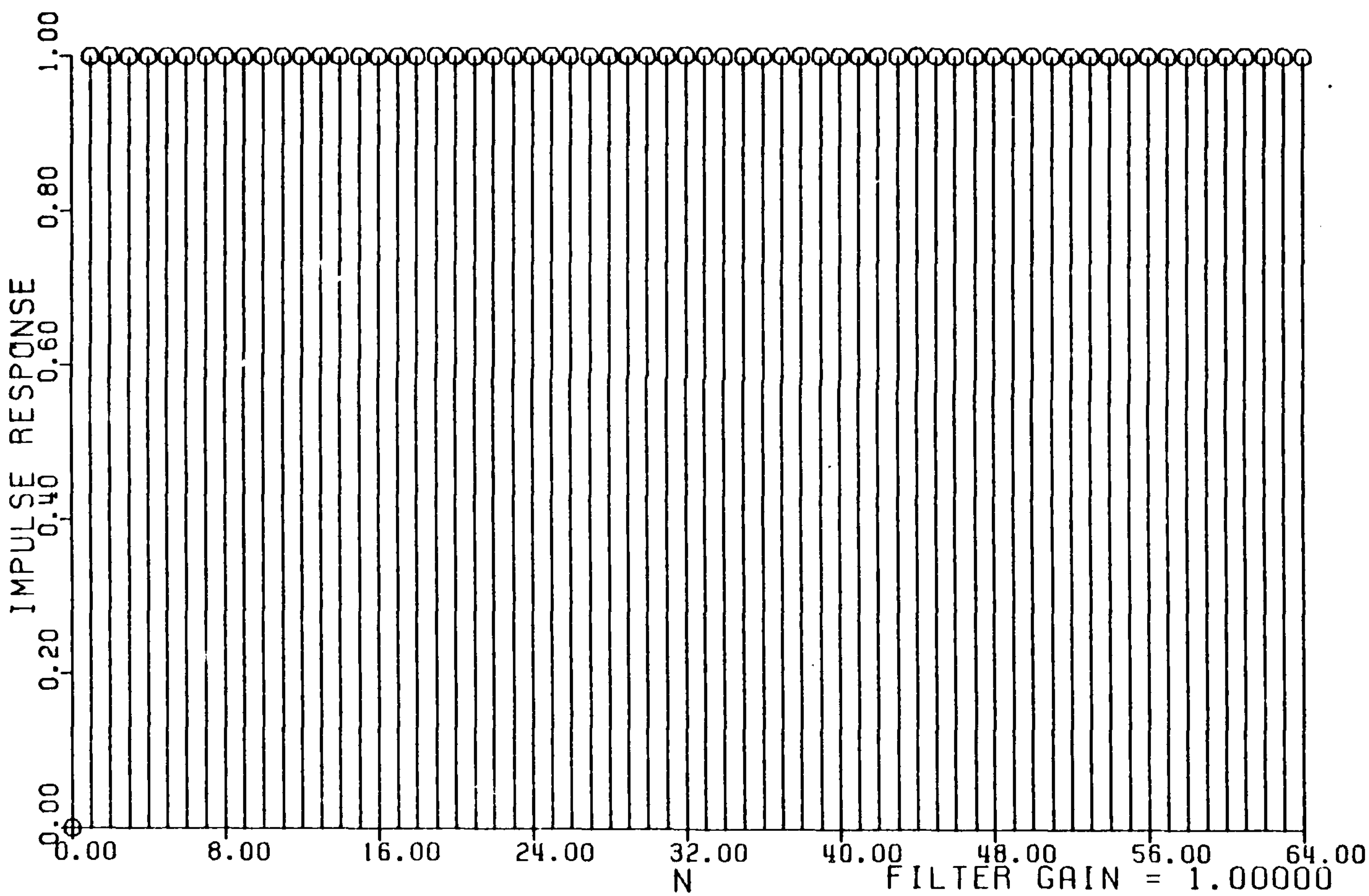


Figure 7-10b Unit Sample Response For A Single Real Pole At (1.0)

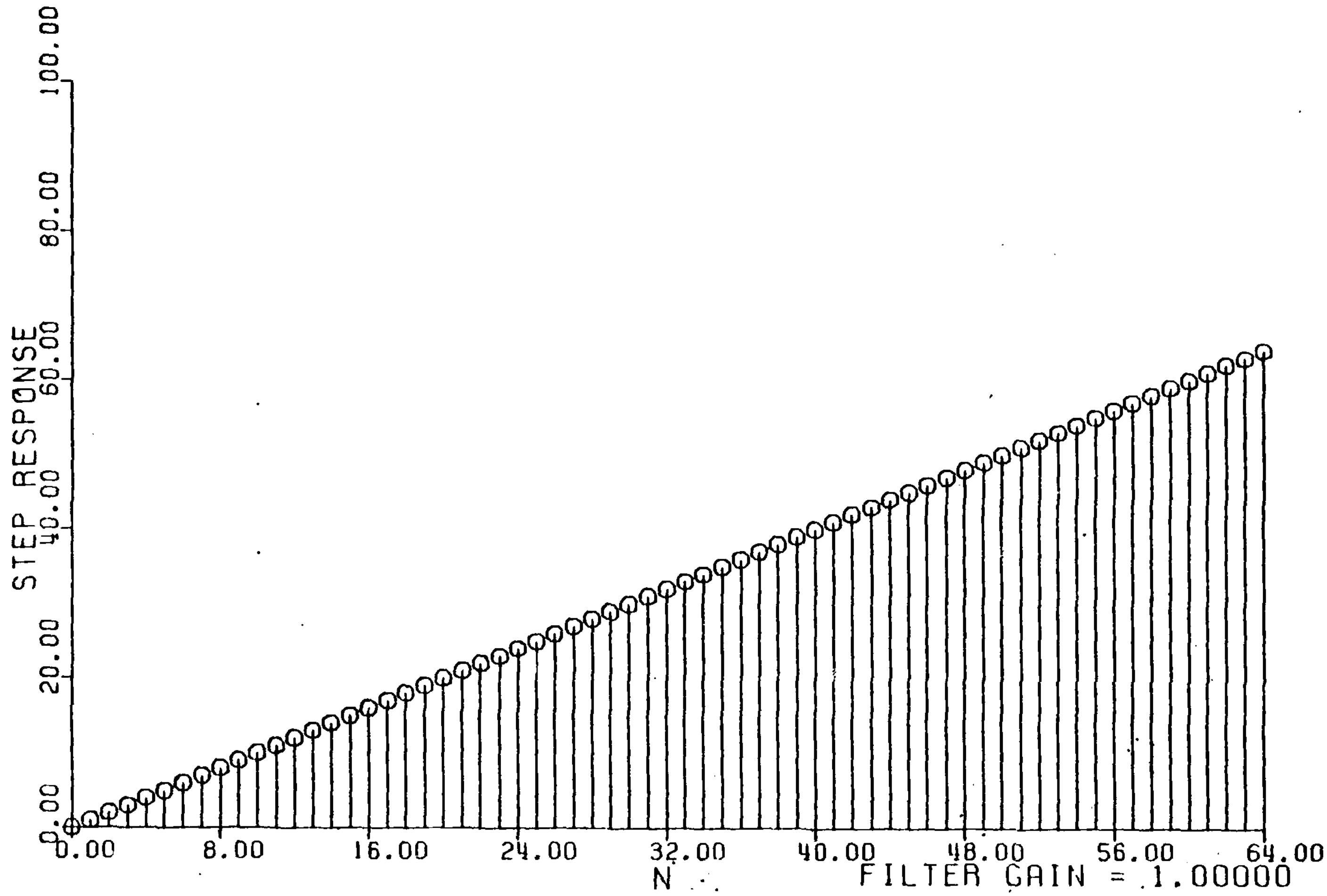


Figure 7-10c Unit Step Response For A Single Real Pole At (1.0)

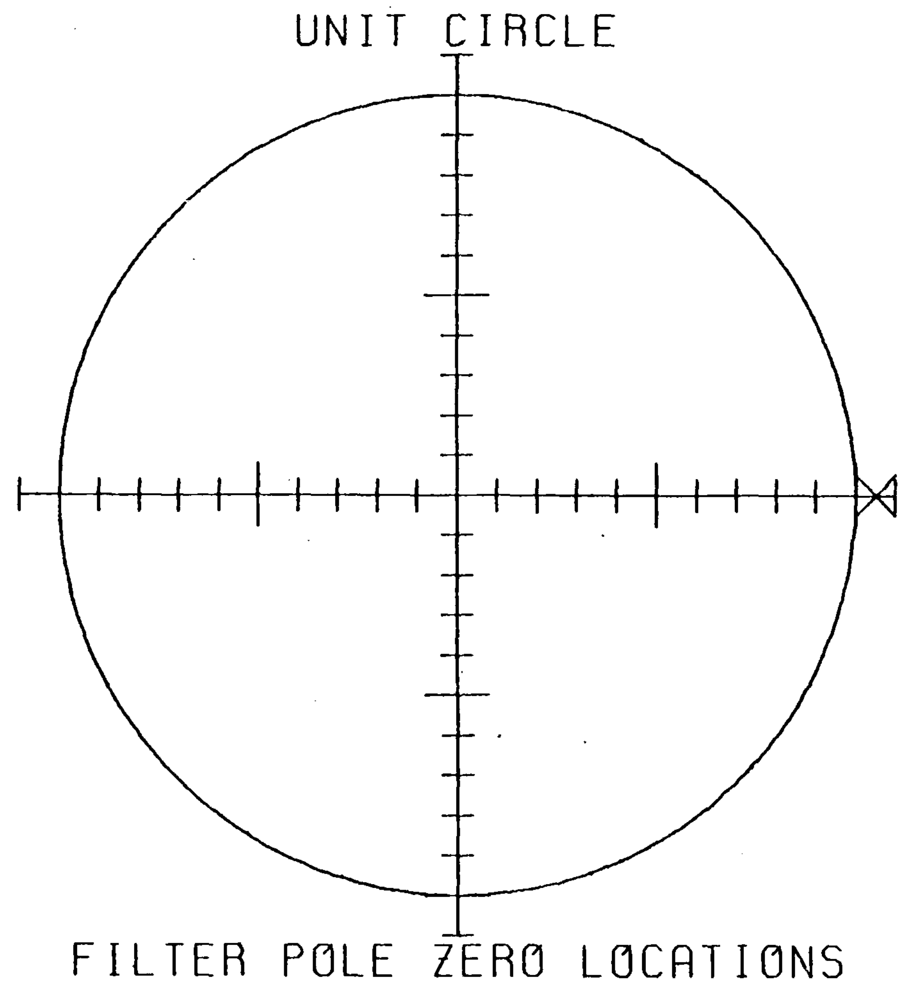


Figure 7-11a Example Of A Single Real Pole Outside The Unit Circle At (1.05)

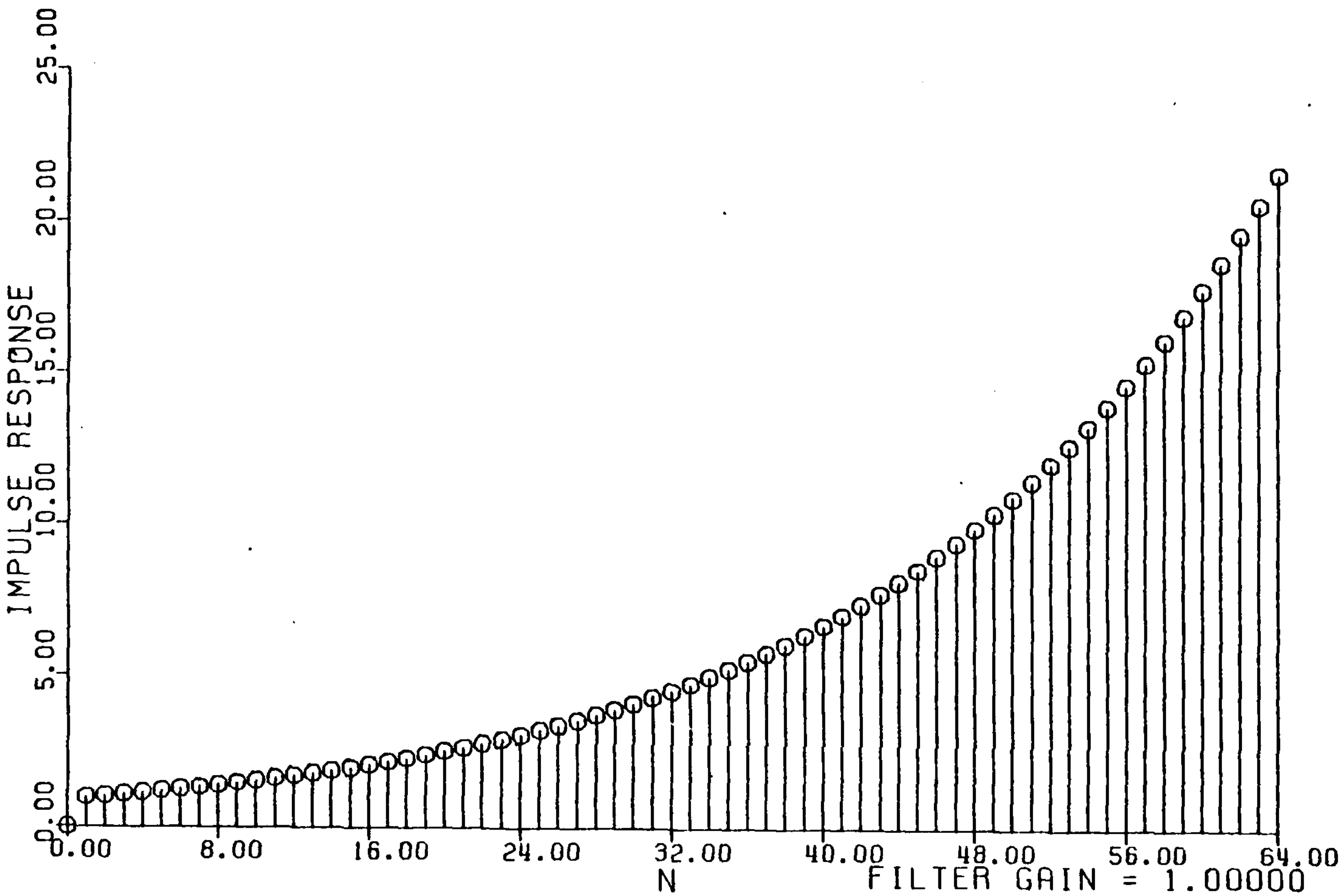


Figure 7-11b Unit Sample Response For A Single Real Pole At (1.05)

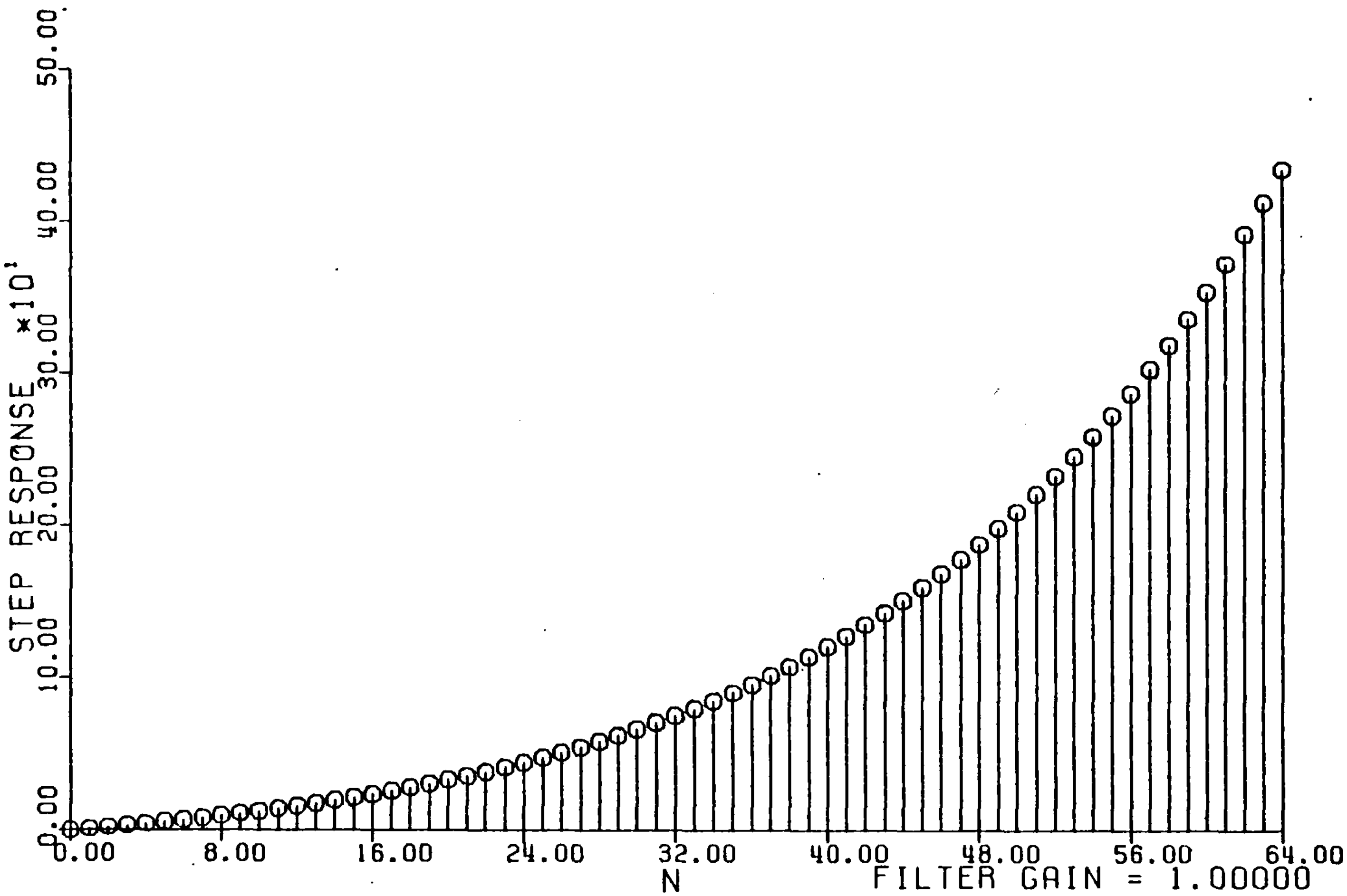


Figure 7-11c Unit Step Response For A Single Real Pole At (1.05)

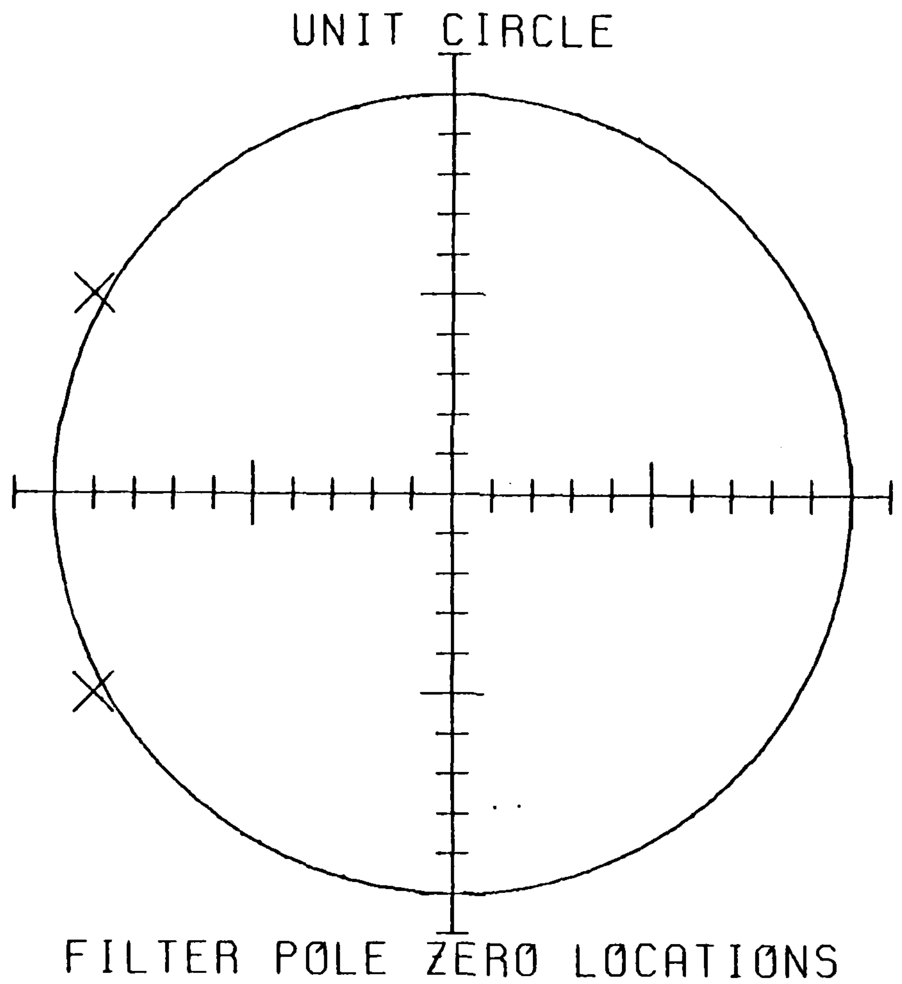


Figure 7-12a Example Of A Complex Pole Pair At $(-0.8 \pm j0.5)$

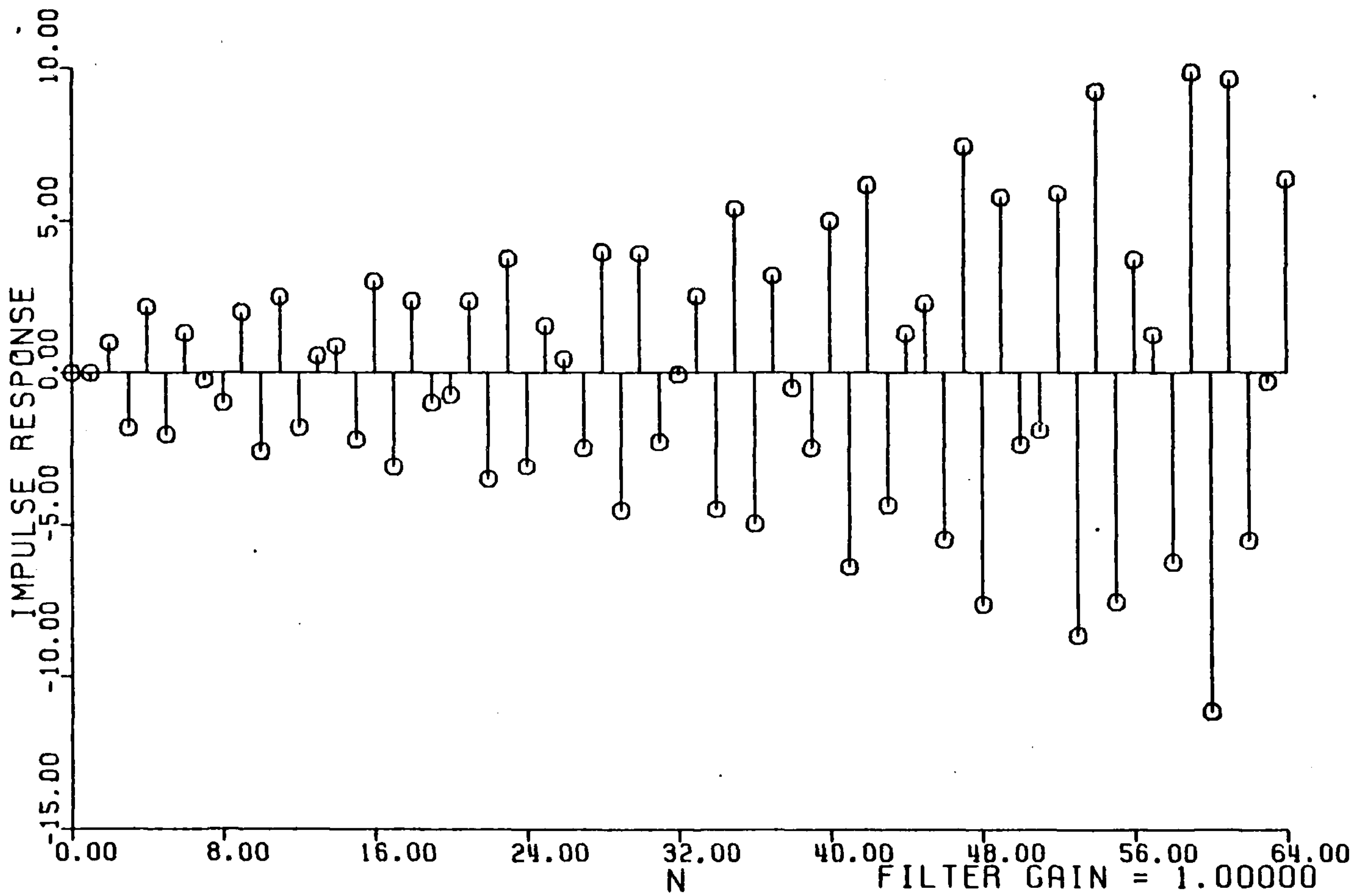


Figure 7-12b Unit Sample Response For A Complex Pole Pair At (-0.8 ±j0.5)

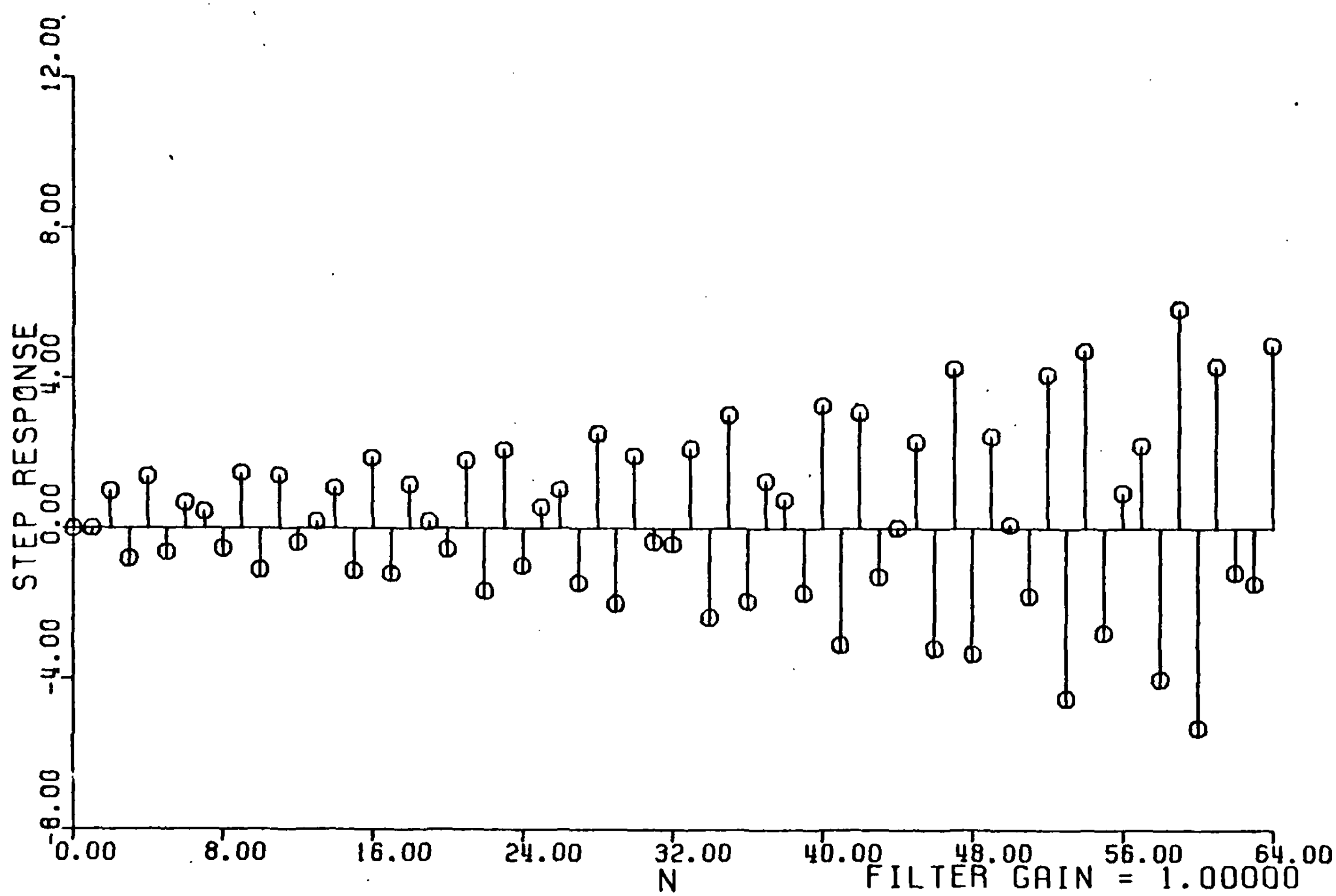


Figure 7-12c Unit Step Response For A Complex Pole Pair At (0.8 ±j0.5)

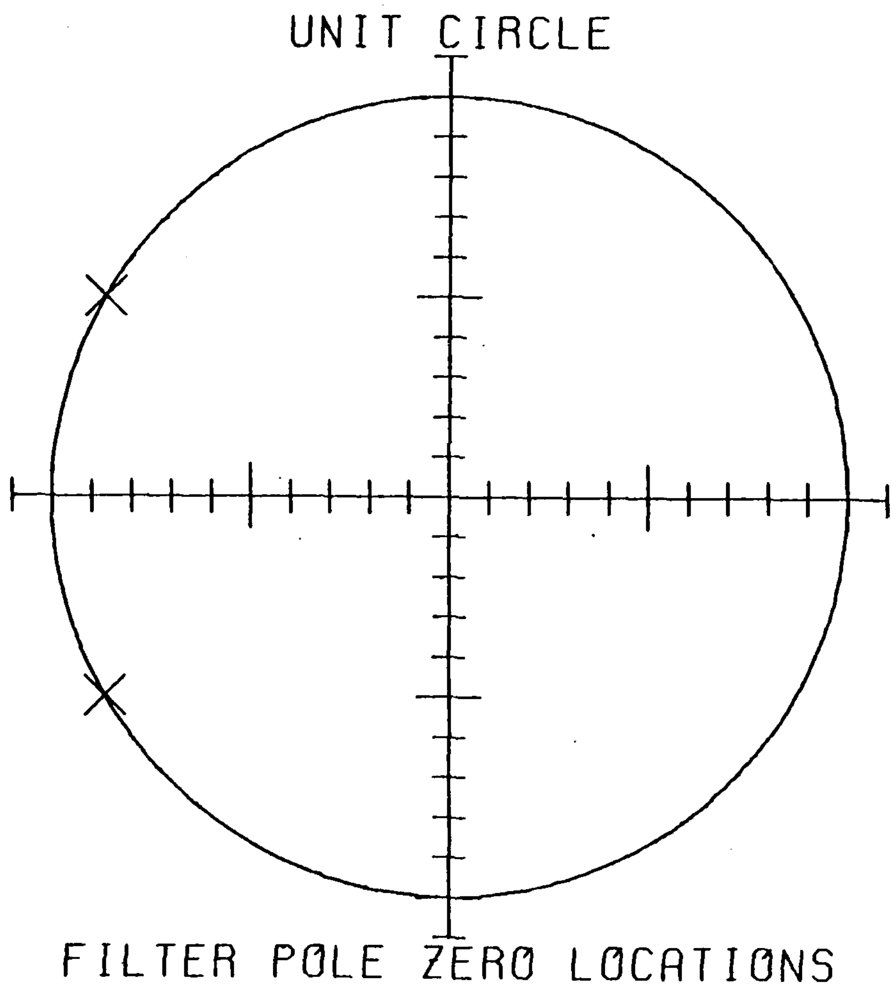


Figure 7-13a Example Of A Complex Pole Pair On The Unit Circle At $(-.866025 \pm j0.5)$

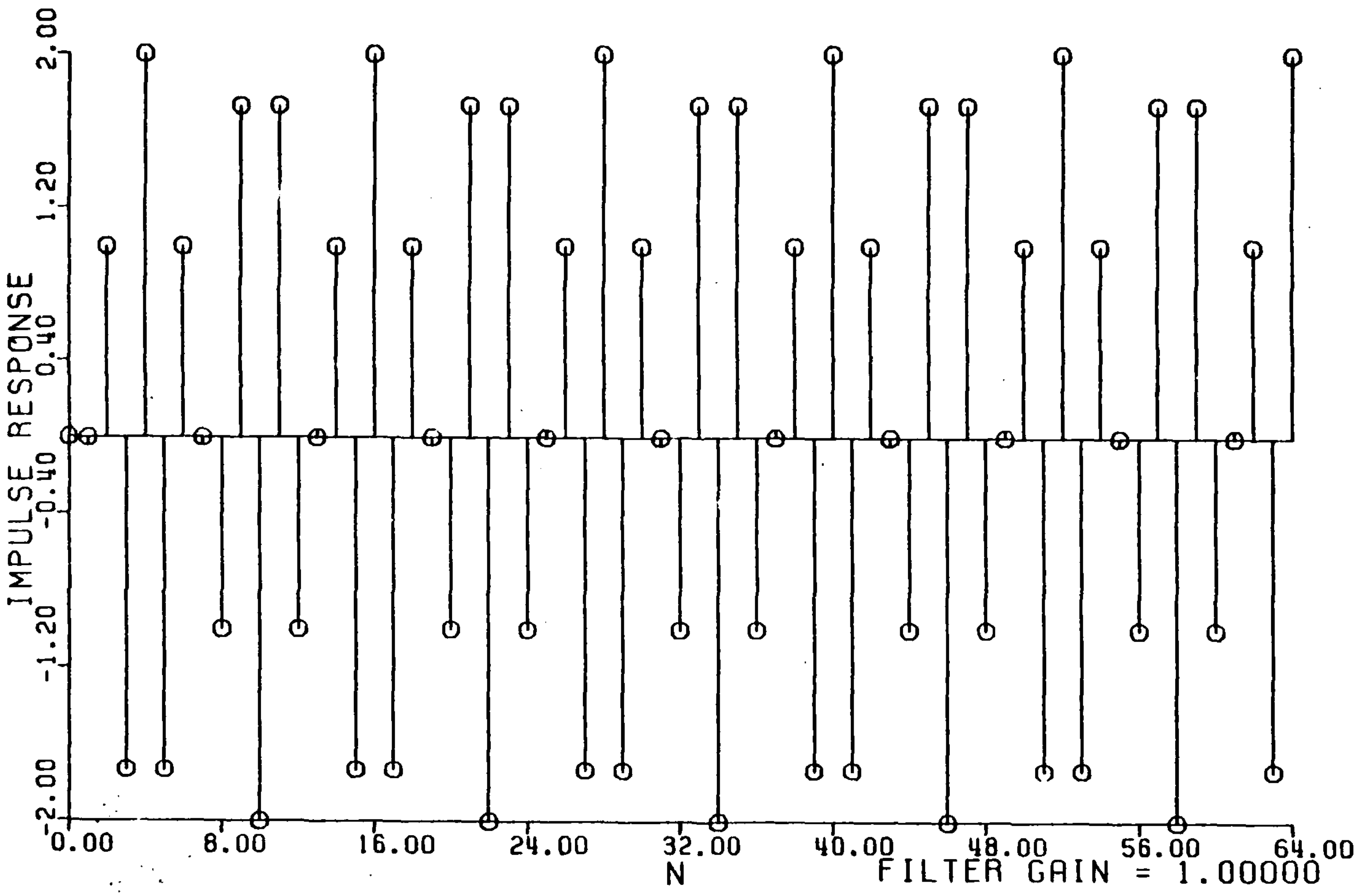


Figure 7-13b Unit Sample Response Of A Complex Pole Pair At (-.866025 $\pm$j0.5)

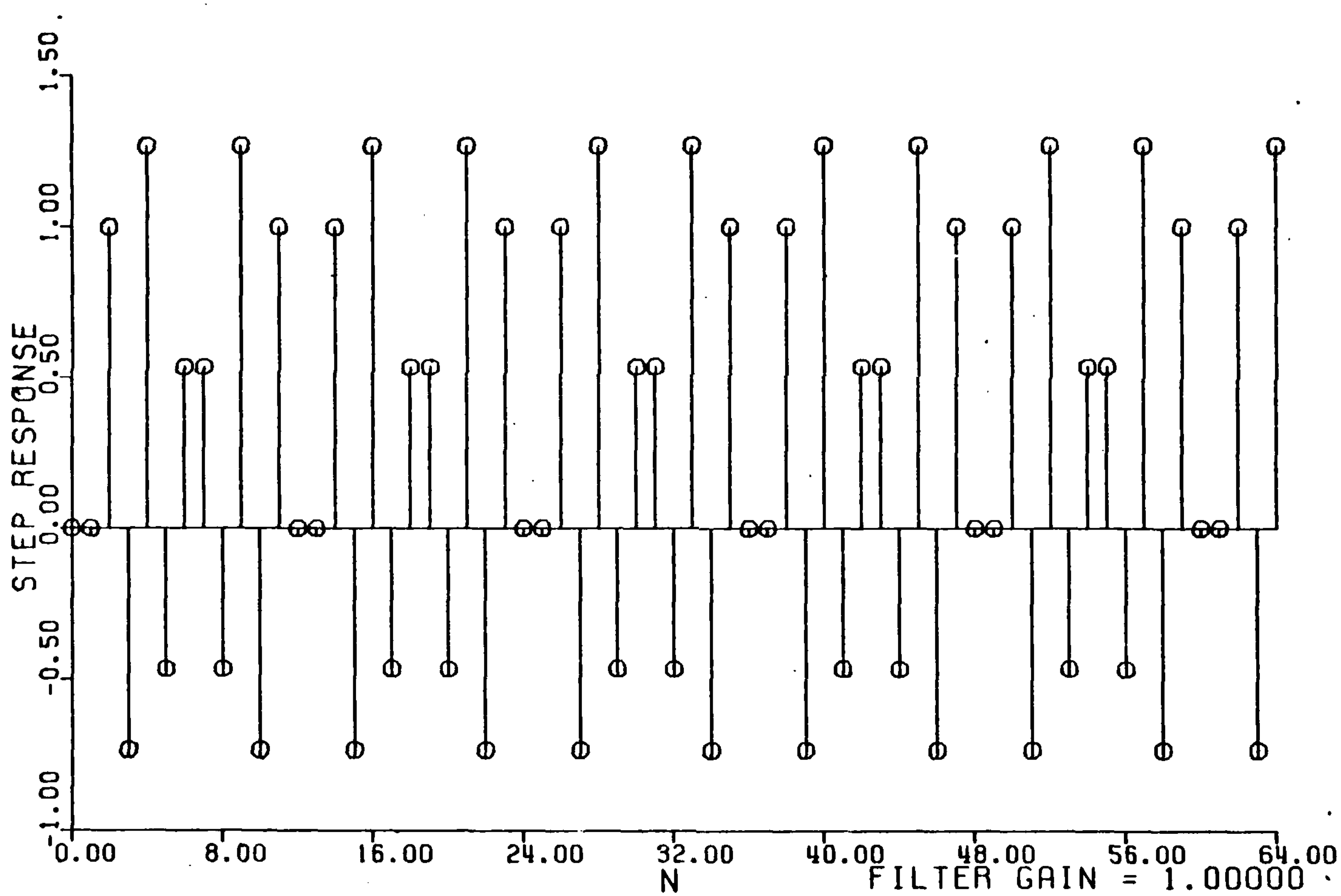


Figure 7-13c Unit Step Response Of A Complex Pole Pair At (-0.866025 $\pm$j0.5)

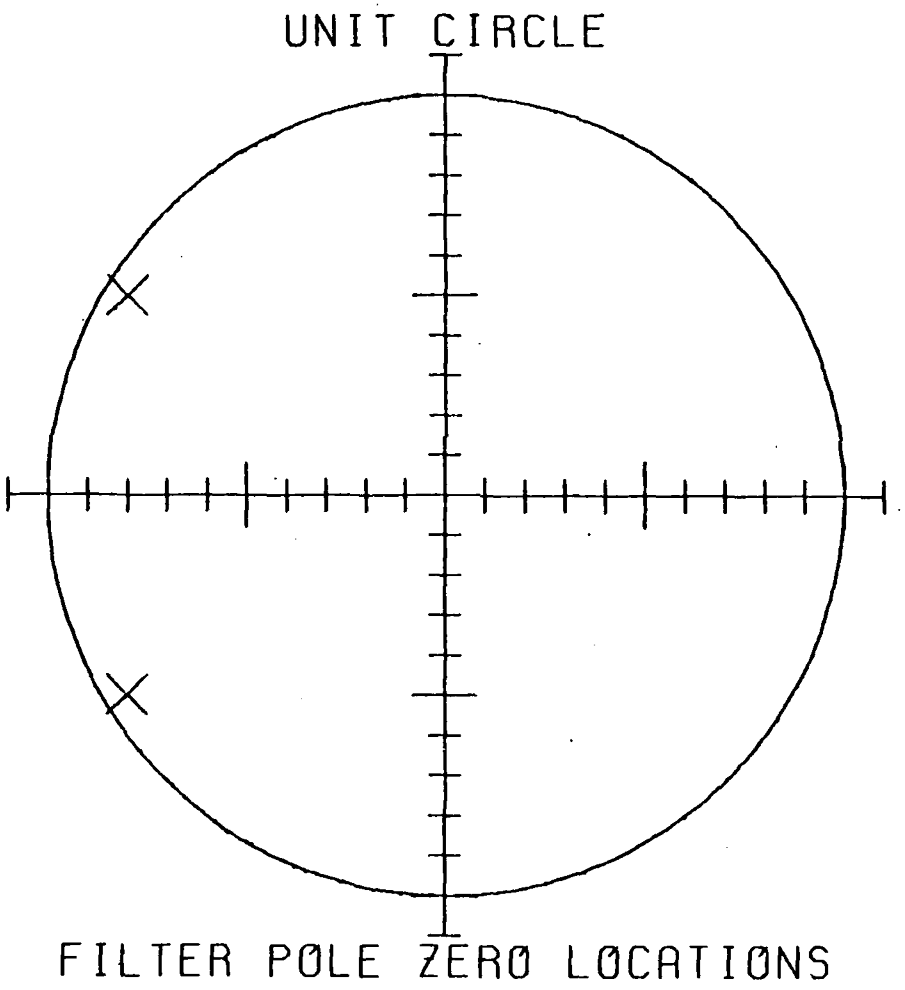


Figure 7-14a Example Of A Complex Pole Pair At ($-0.8 \pm j0.5$)

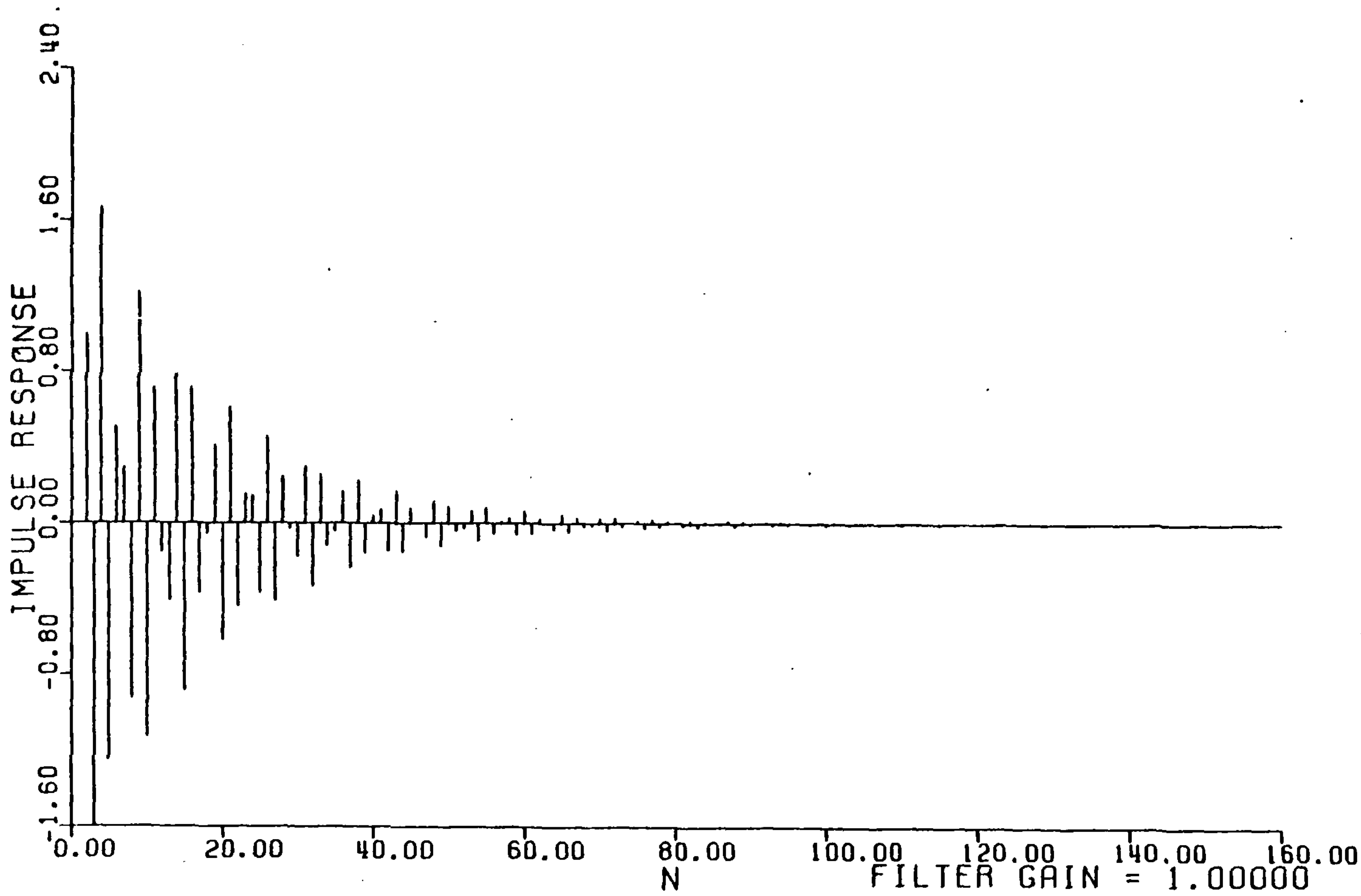


Figure 7-14b Unit Sample Response For A Complex Pole Pair At (-0.8 ±j0.5)

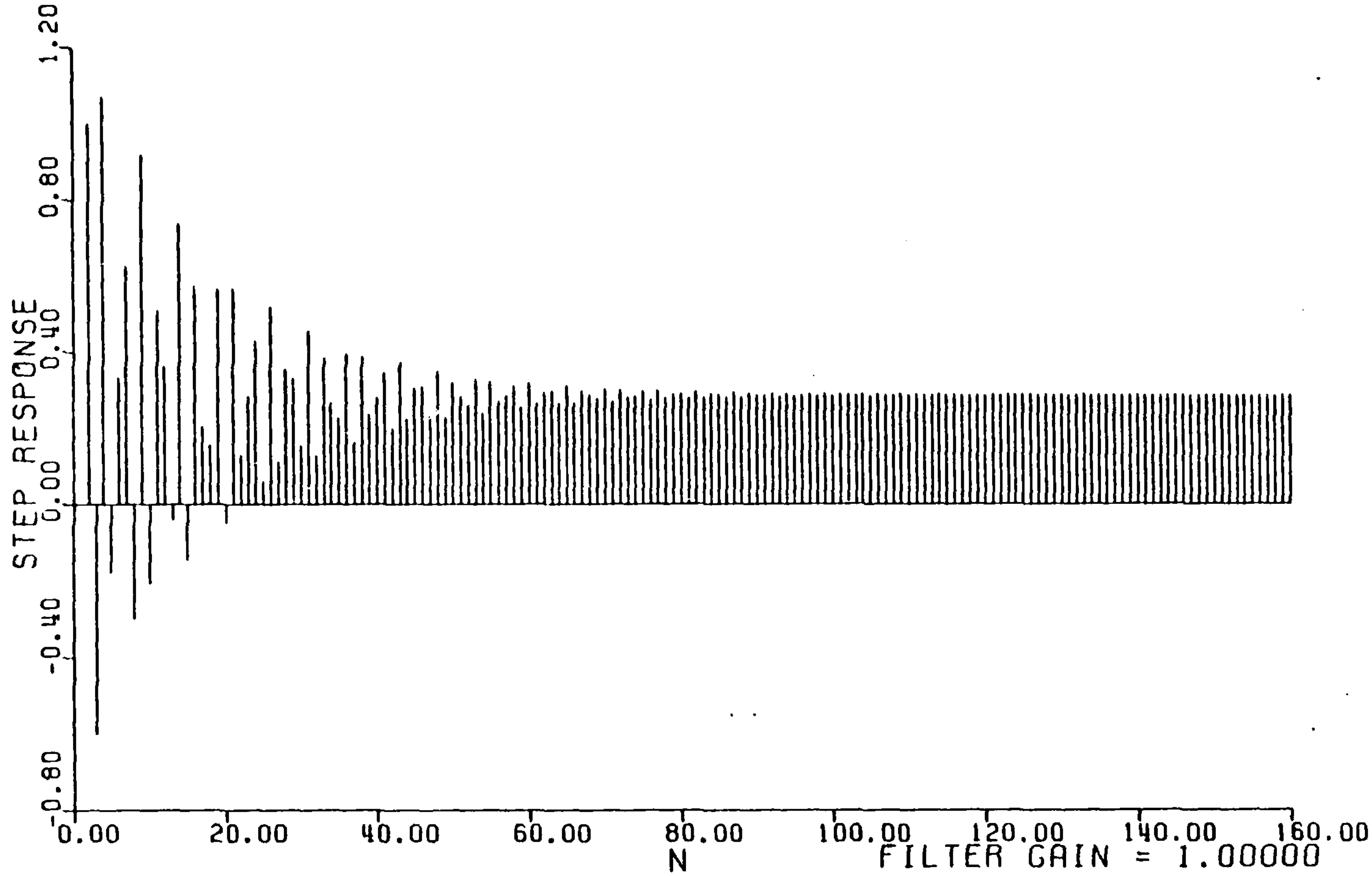


Figure 7.14c Unit Step Response For a Complex Pole Pair At (-0.8 $\pm j0.5$)

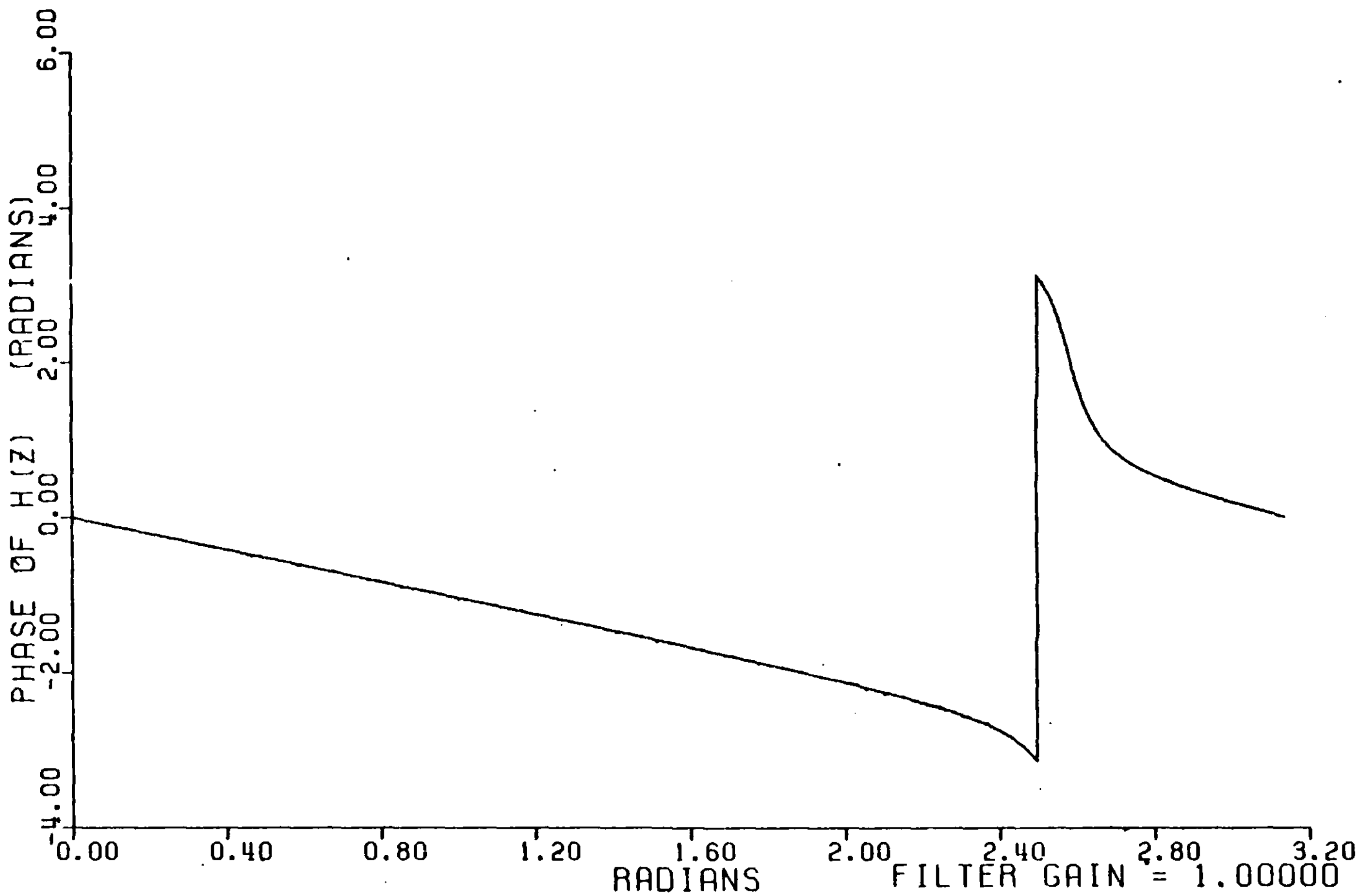


Figure 7-14d Phase Response For A Complex Pole Pair At (-0.8 ±j0.5)

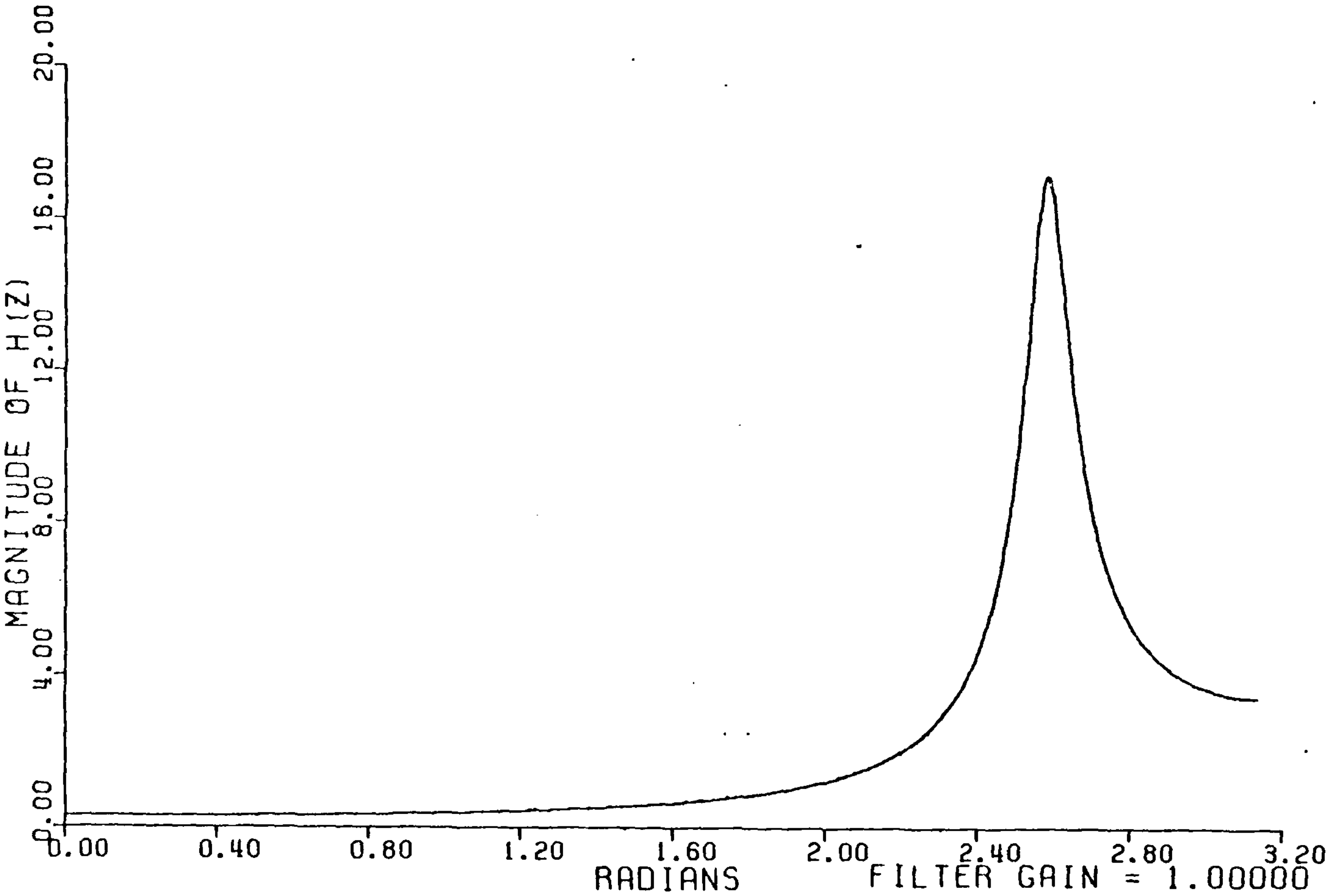


Figure 7-14e Magnitude Of H(z) For A Complex Pole Pair At (-0.8 $\pm$j0.5)

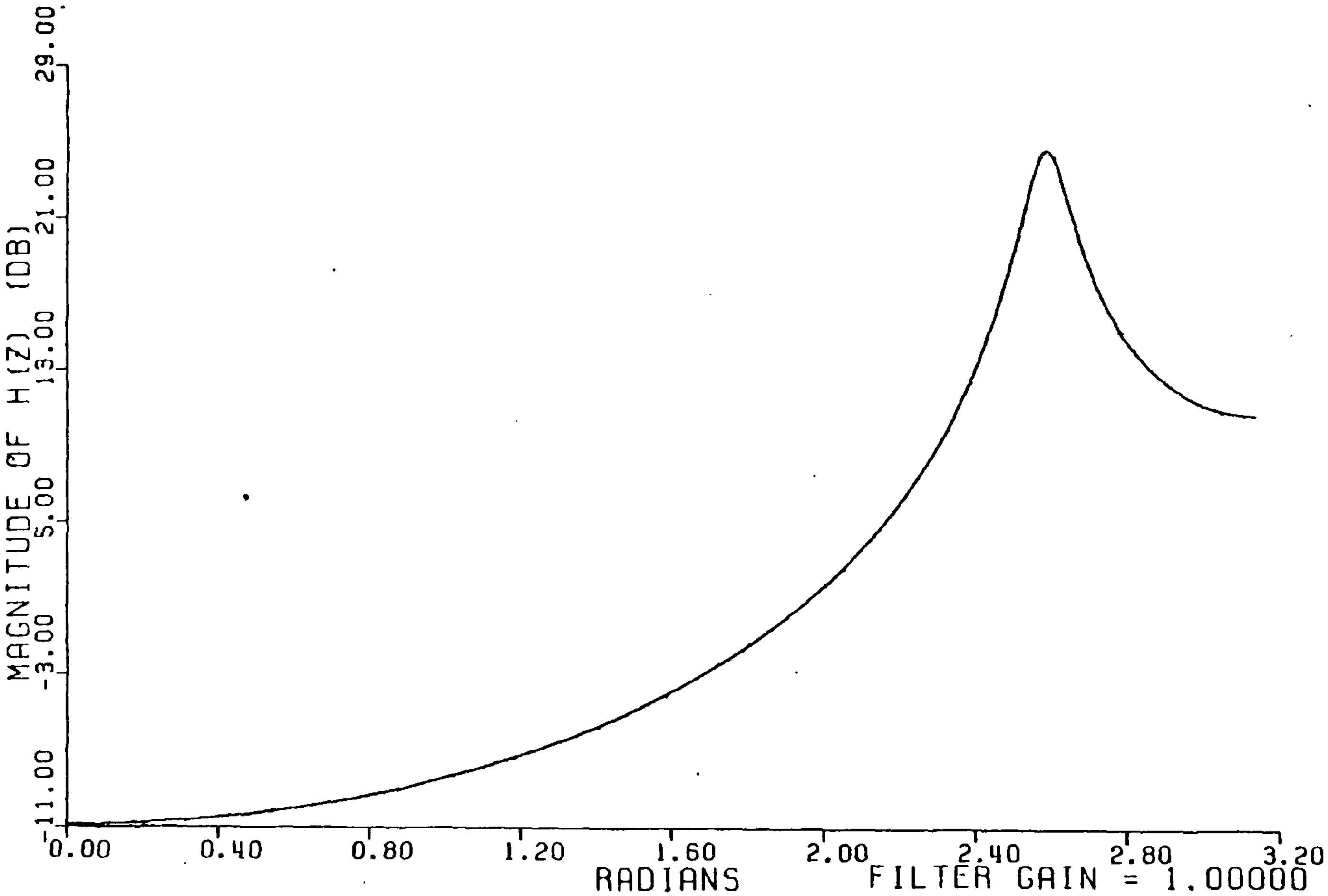


Figure 7-14f Magnitude Of H(z) In Decibels For A Complex Pole Pair At (-0.8 $\pm$j0.5)

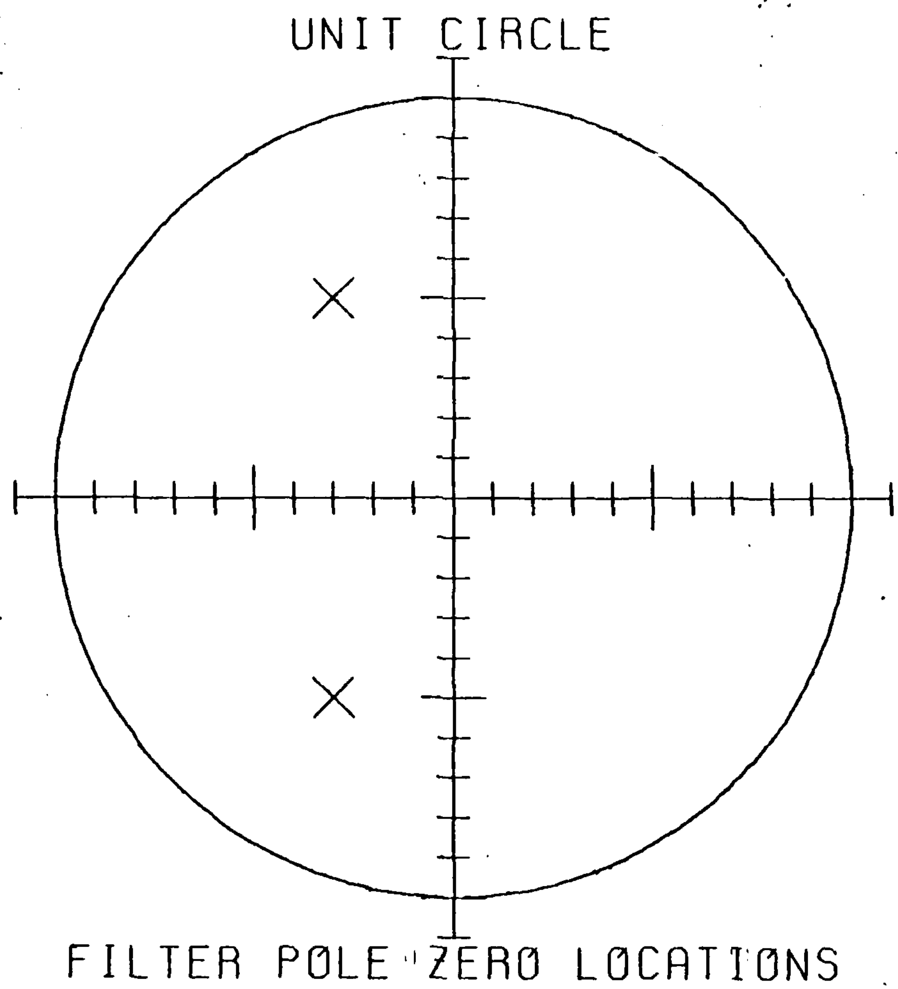


Figure 7-15a Example Of A Complex Pole Pair At (-0.3 $\pm j0.5$)

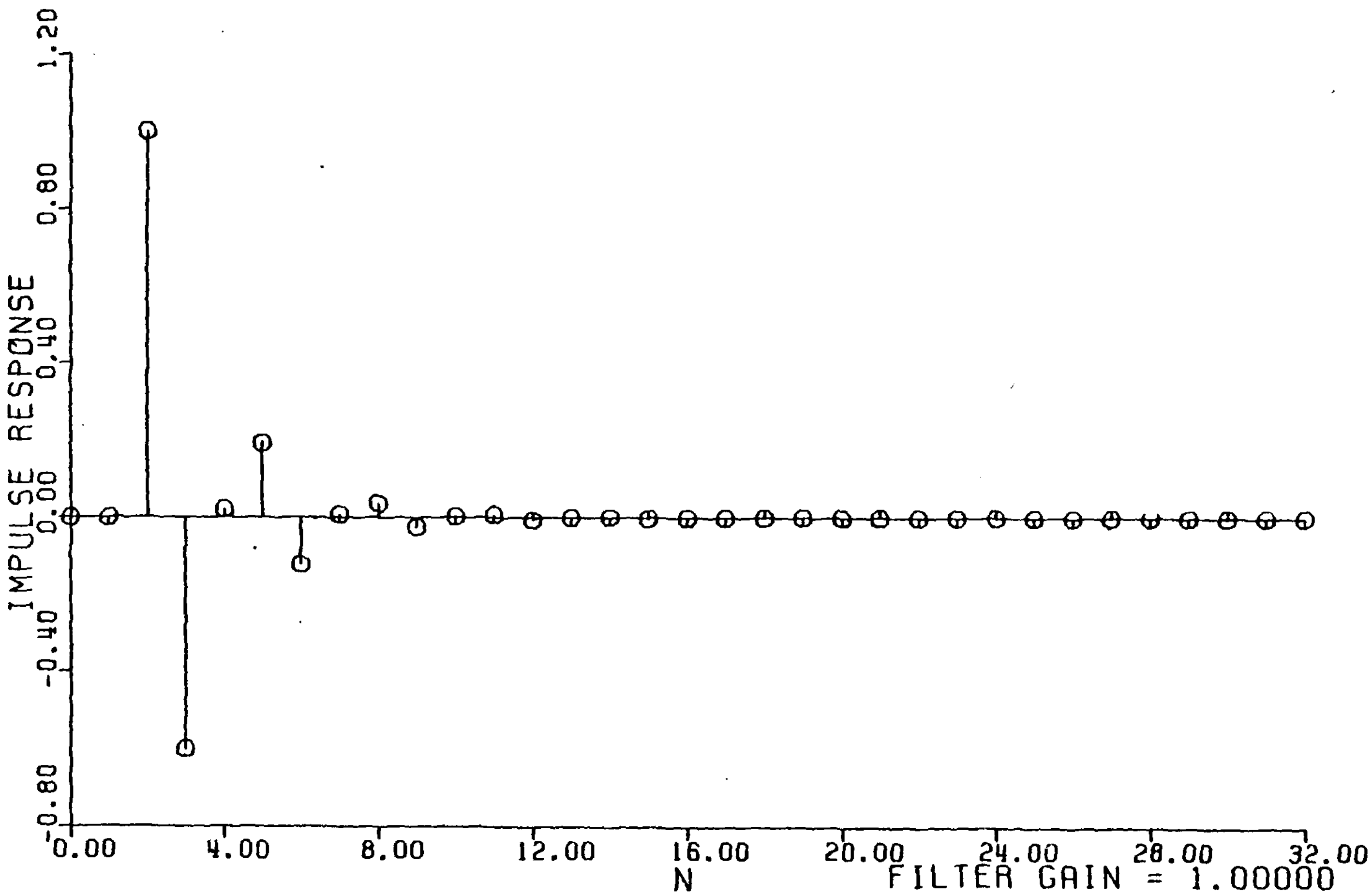


Figure 7-15b Unit Sample Response For A Complex Pole Pair At (-0.3 $\pm j0.5$)

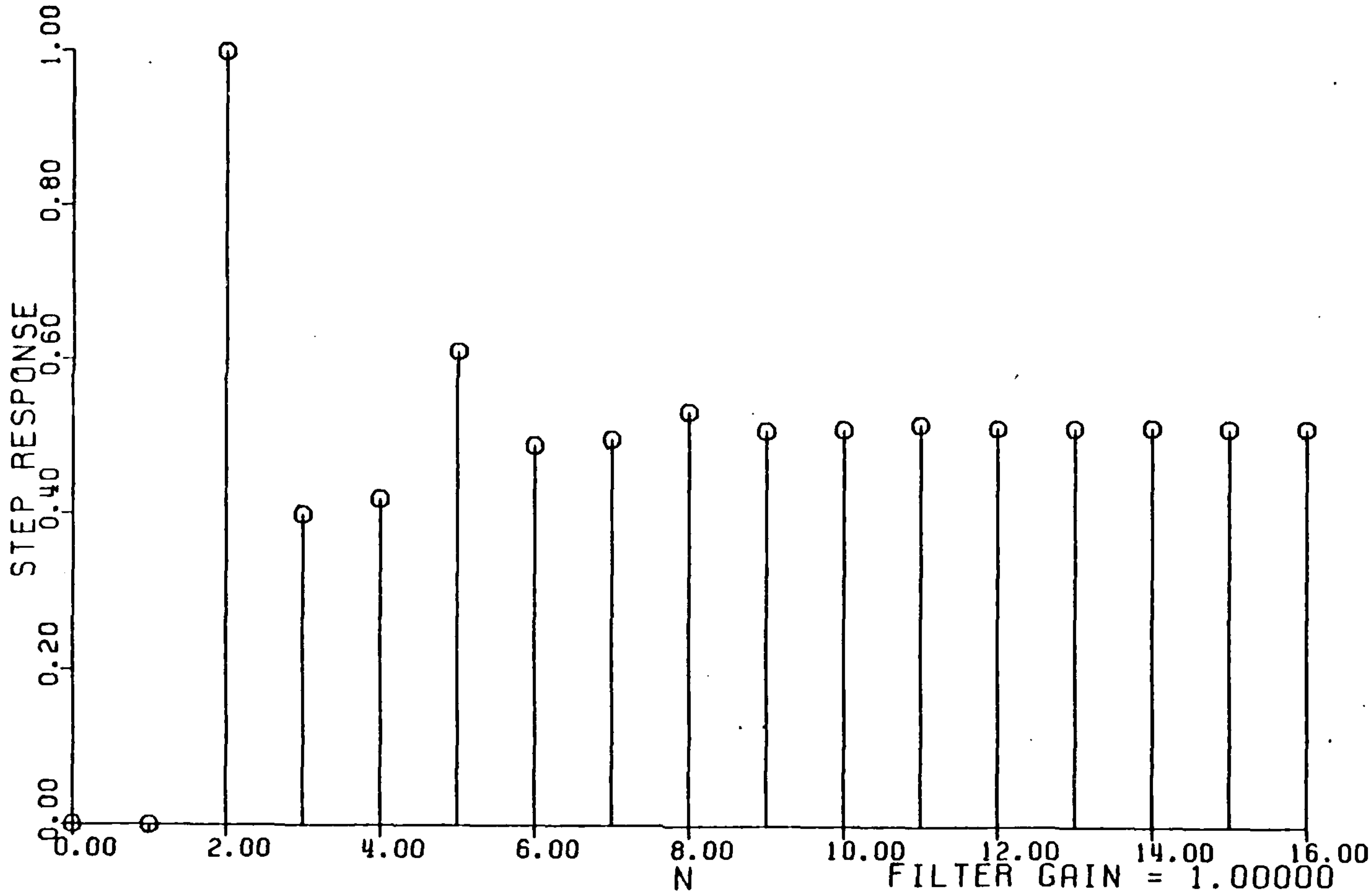


Figure 7-15c Unit Step Response For A Complex Pole Pair At (-0.3 $\pm$j0.5)

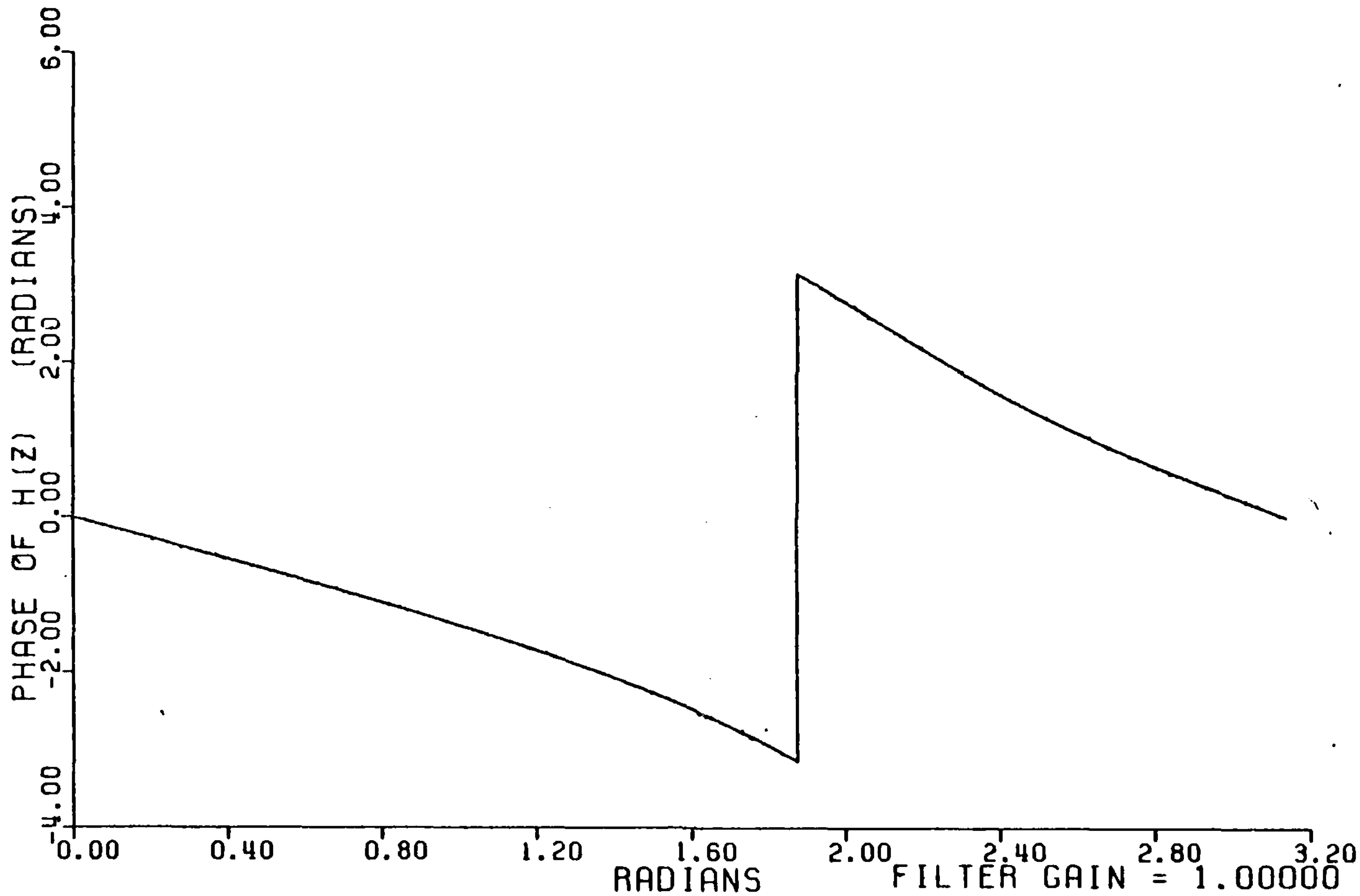


Figure 7-15d Phase Response For A Complex Pole Pair At (-0.3 $\pm$j0.5)

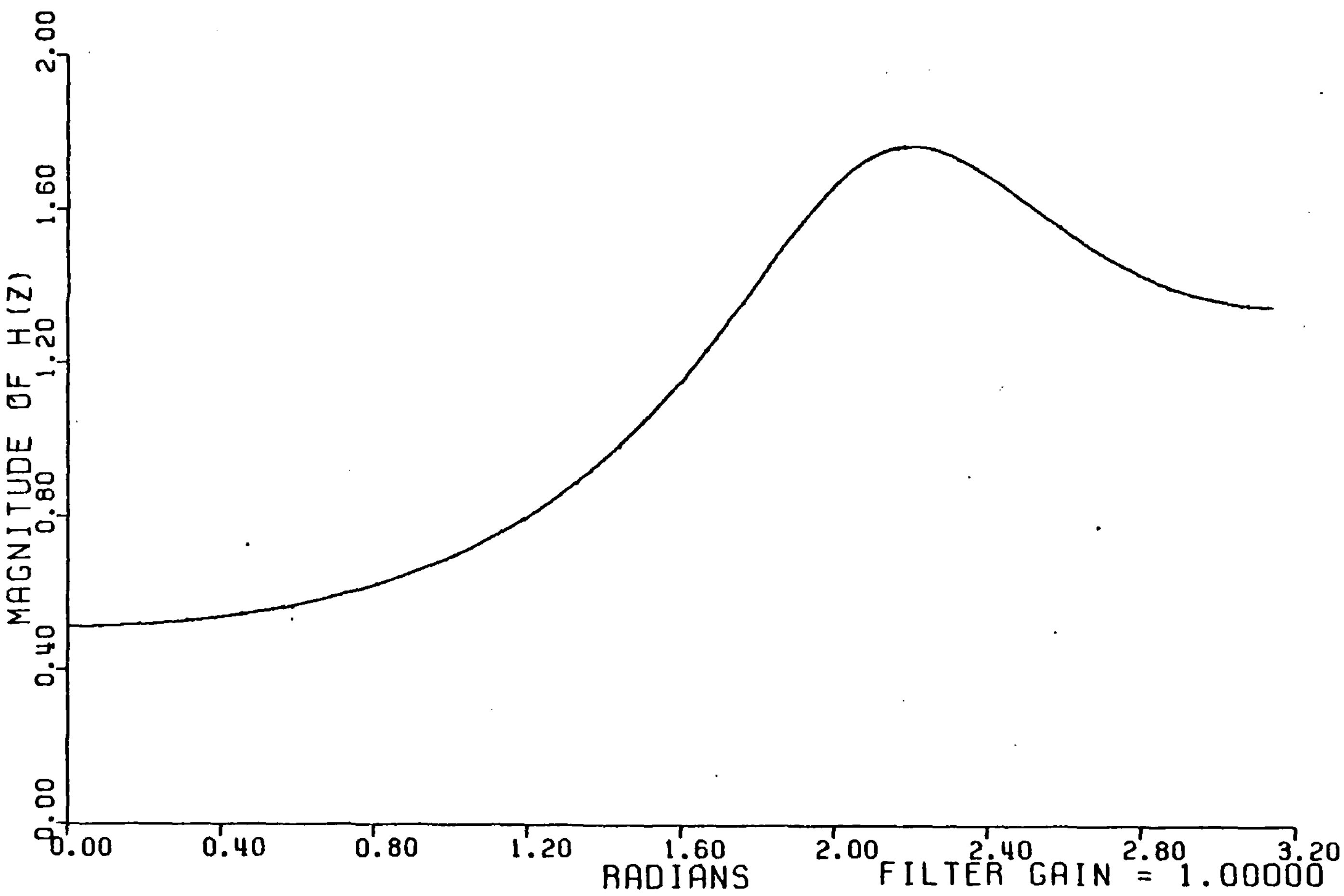


Figure 7-15e Magnitude Of H(z) For A Complex Pole Pair At (-0.3 ±j0.5)

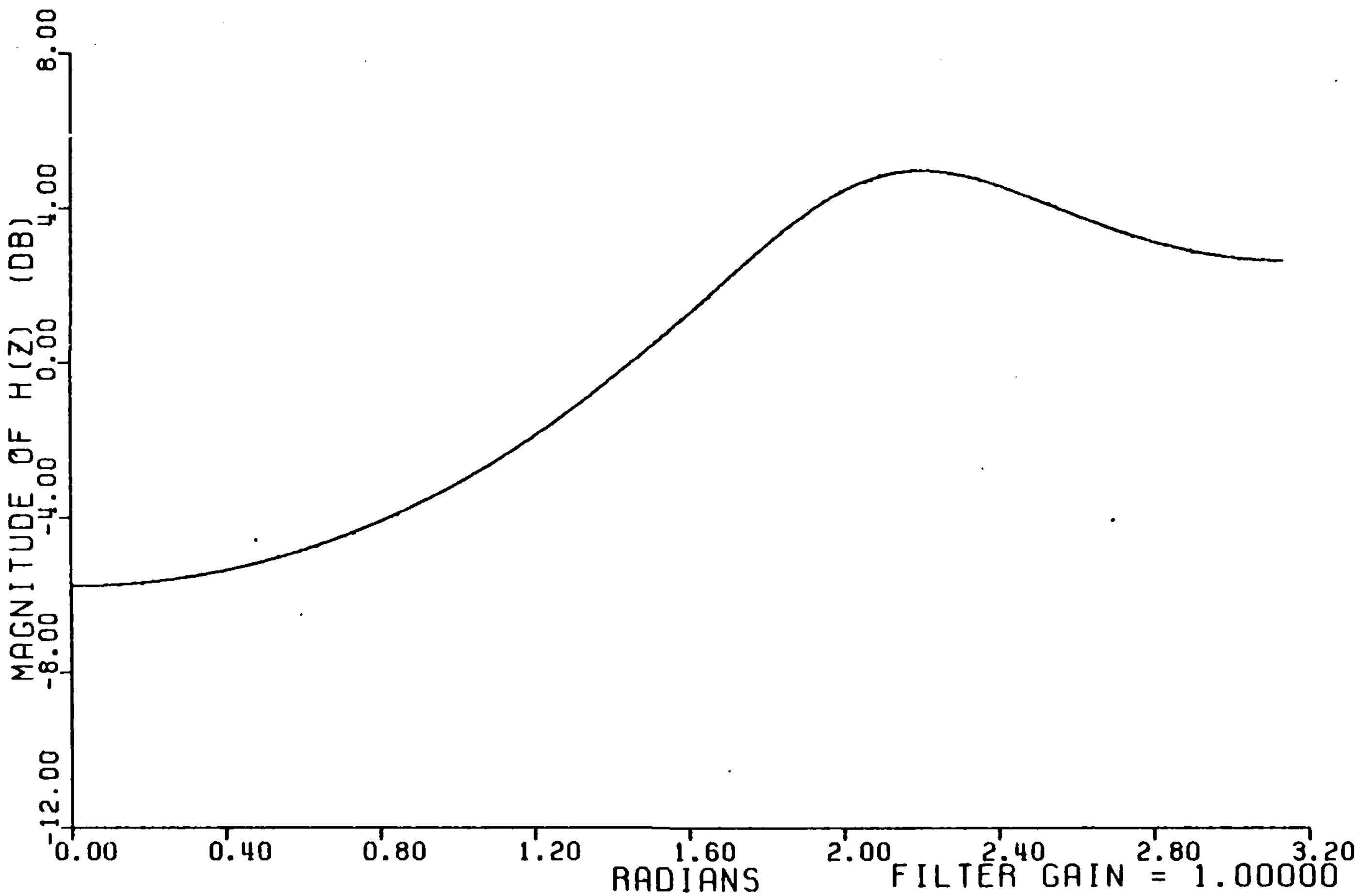


Figure 7-15f Magnitude Of (Hz) In Decibels For A Complex Pole Pair At (-0.3 $\pm j0.5$)

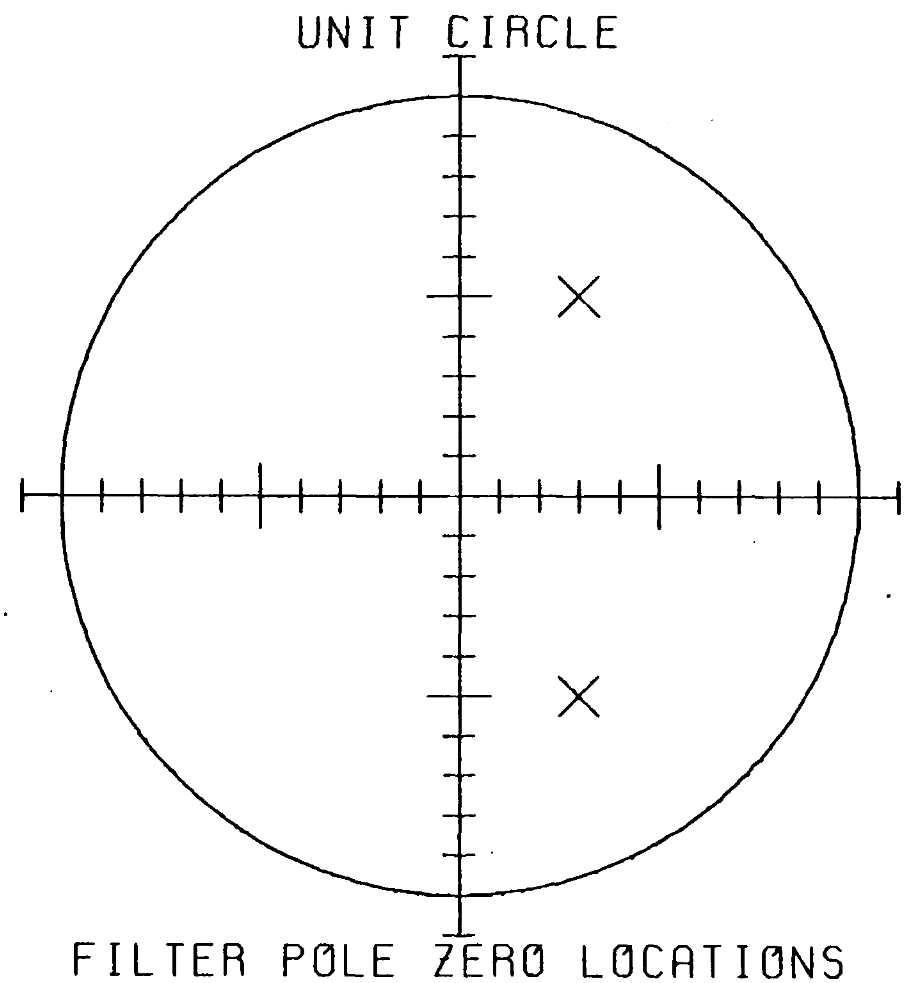


Figure 7-16a Example Of A Complex Pole Pair At (0.3 $\pm$j0.5)

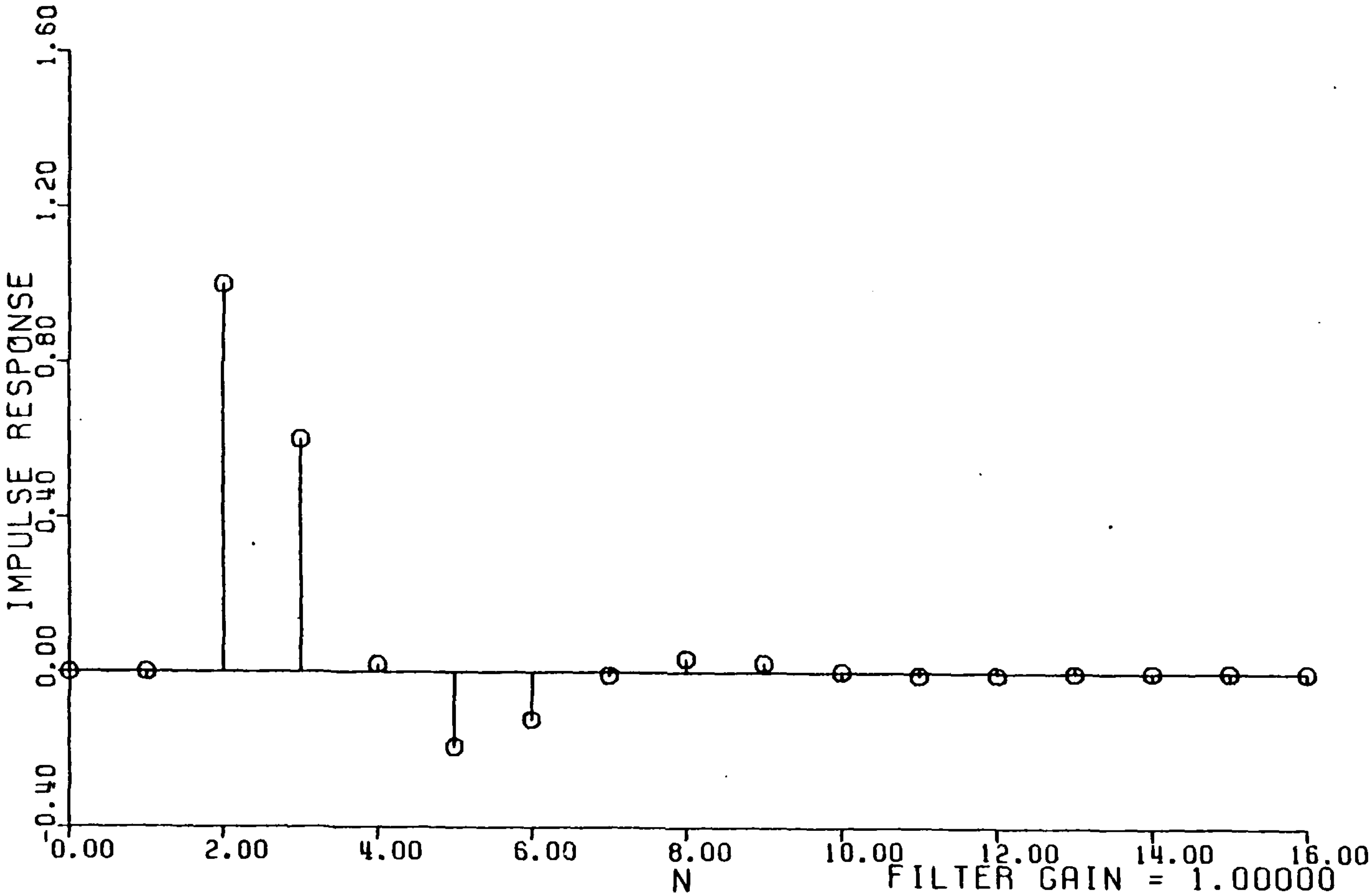


Figure 7-16b Unit Sample Response For A Complex Pole Pair At (0.3 $\pm j0.5$)

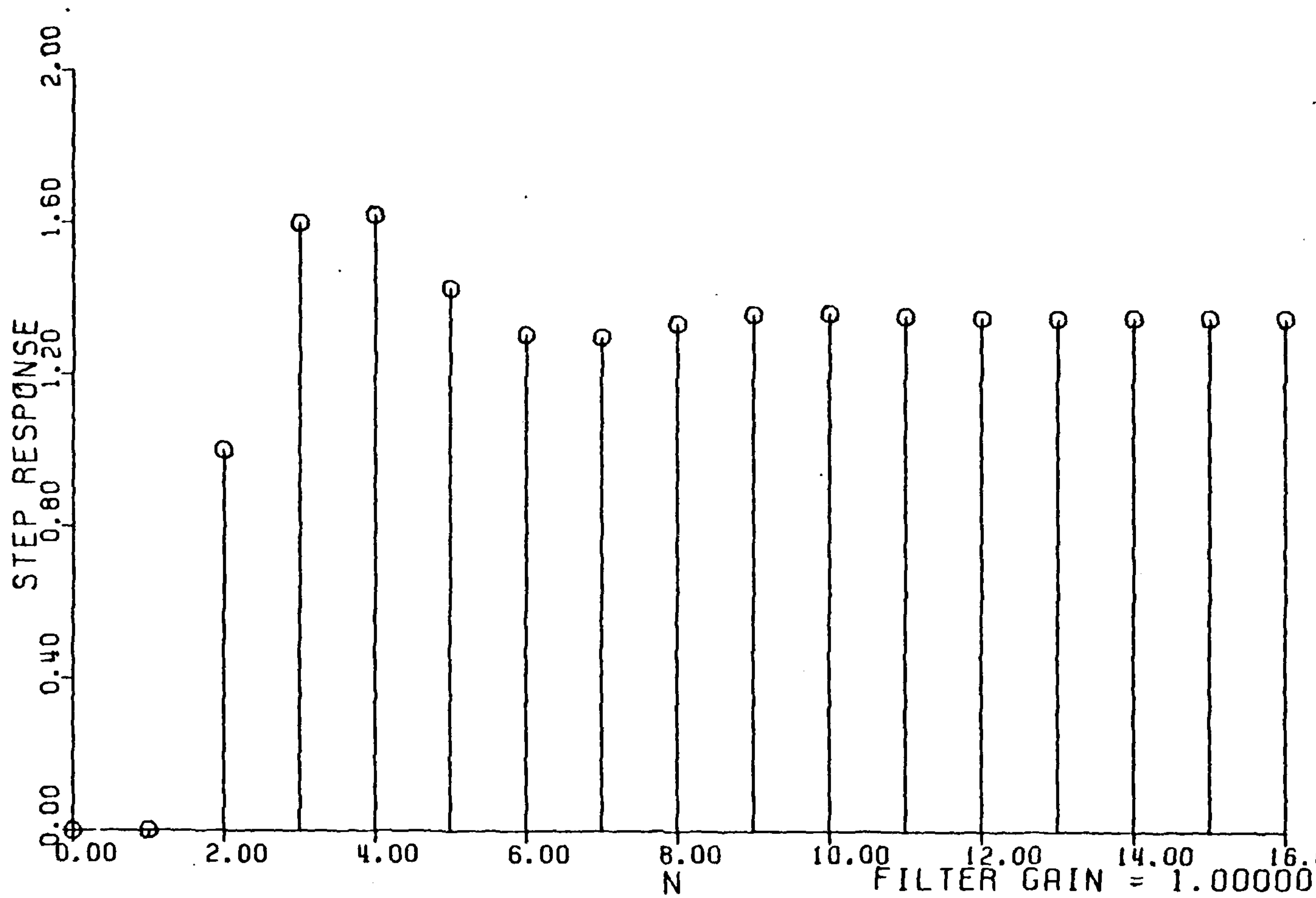


Figure 7-16c Unit Step Response For A Complex Pole Pair At (-0.3 ±j0.5)

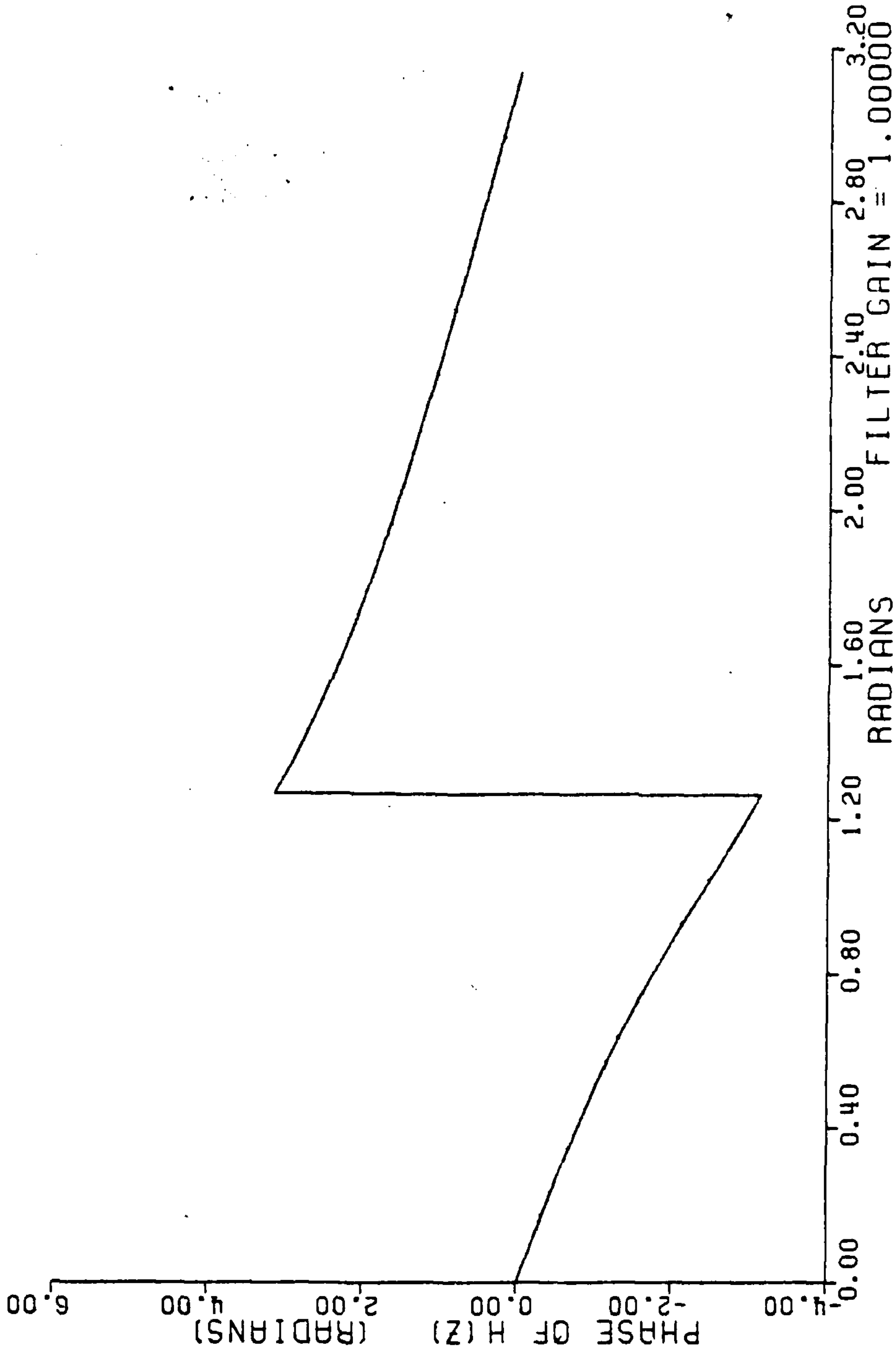


Figure 7-16d Phase Response For A Complex Pole Pair At (0.3 $\pm$j0.5)

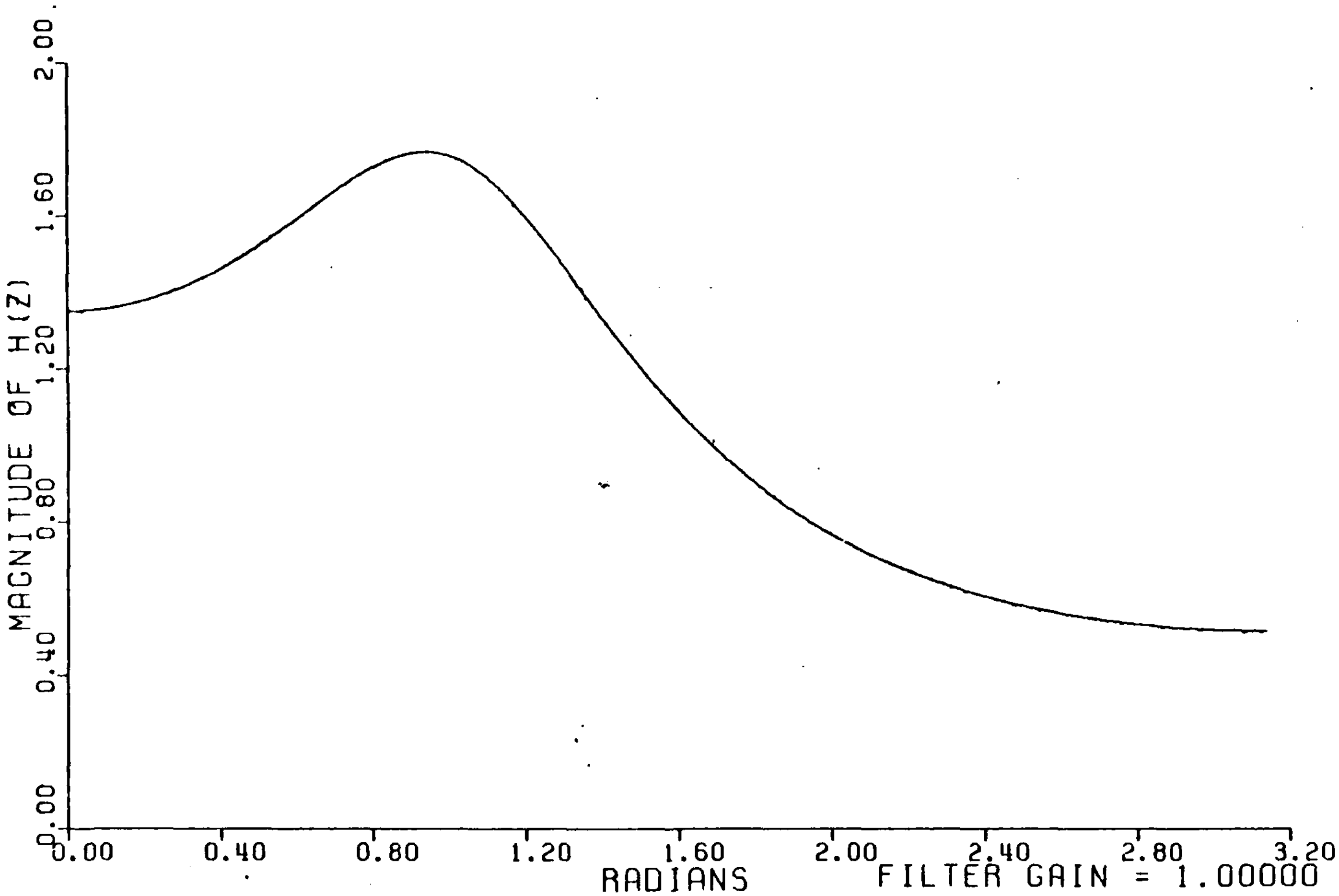


Figure 7-16e Magnitude Of H(z) For A Complex Pole Pair At (0.3 $\pm$j0.5)

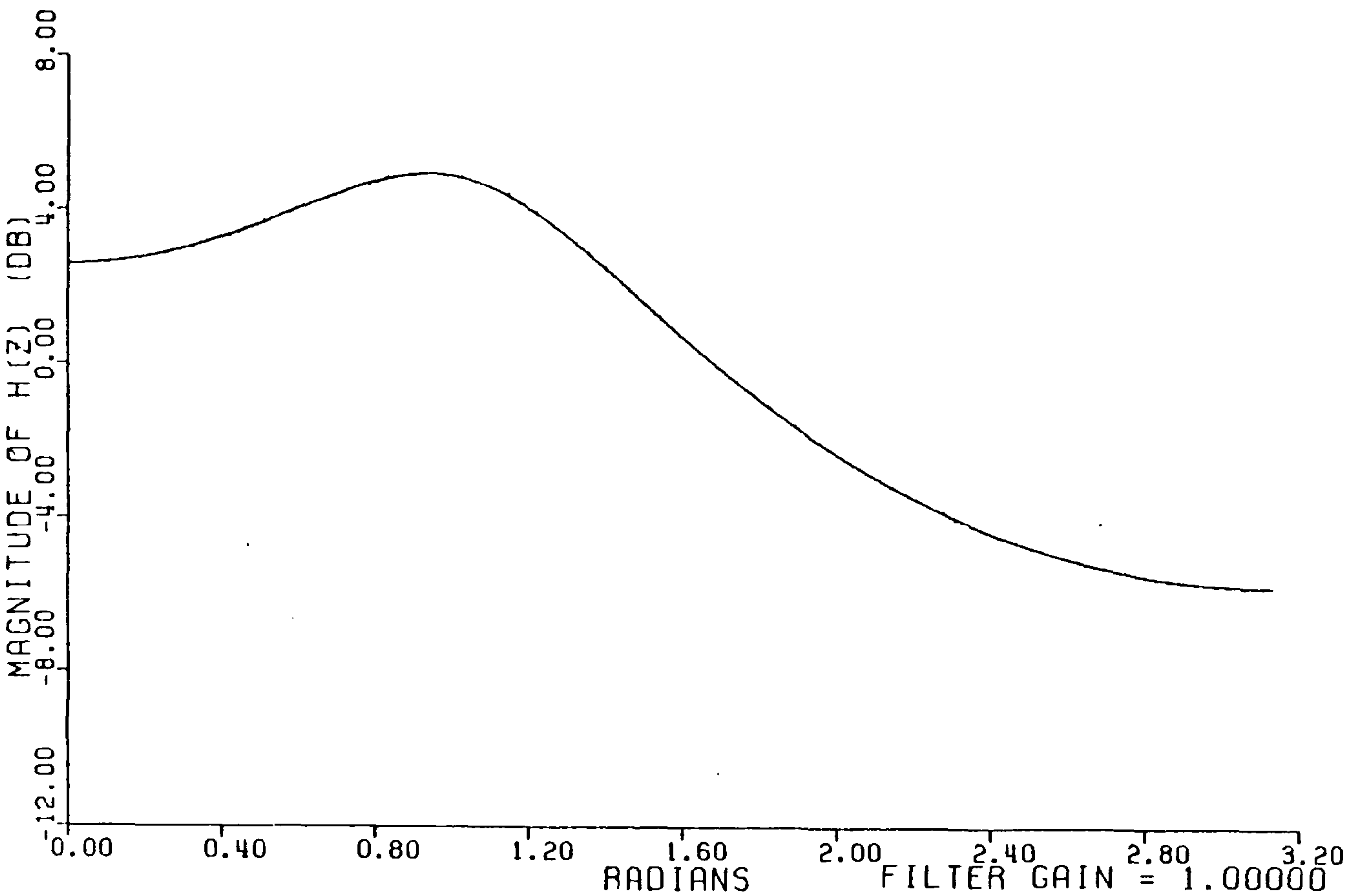


Figure 7-16f Magnitude Of H(z) In Decibels For A Complex Pole Pair At (0.3 $\pm j0.5$)

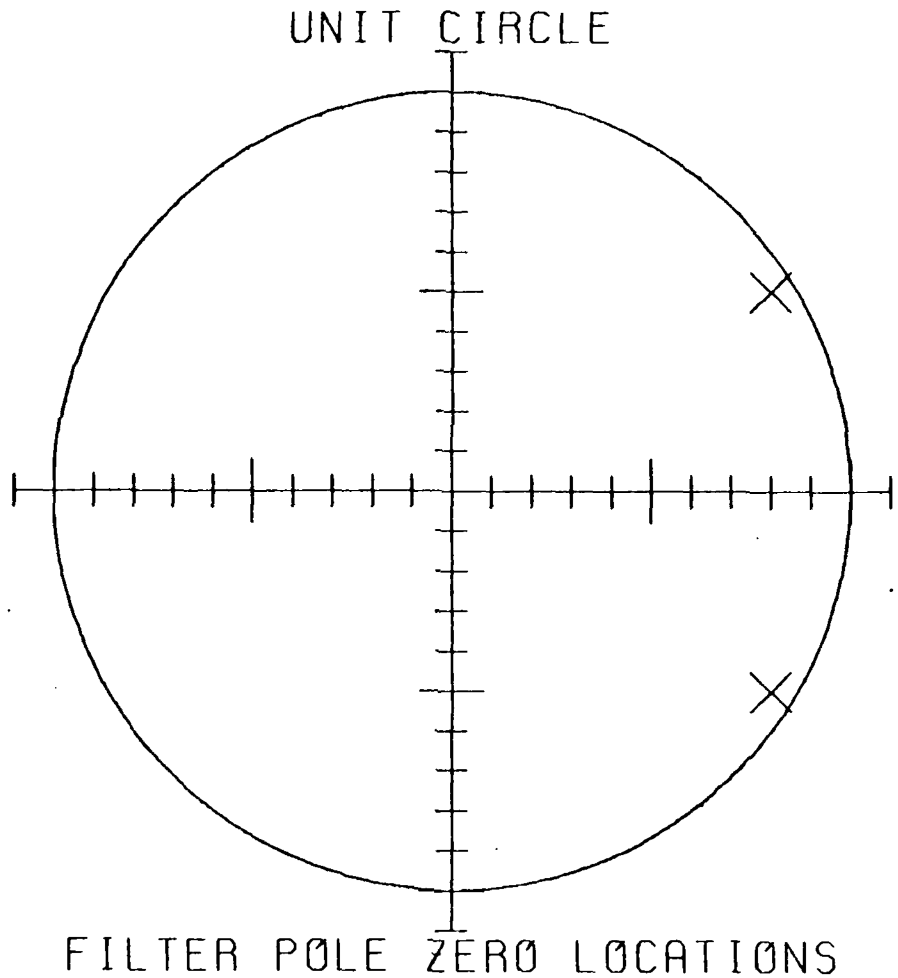


Figure 7-17a Example Of A Complex Pole Pair At $(0.8 \pm j0.5)$

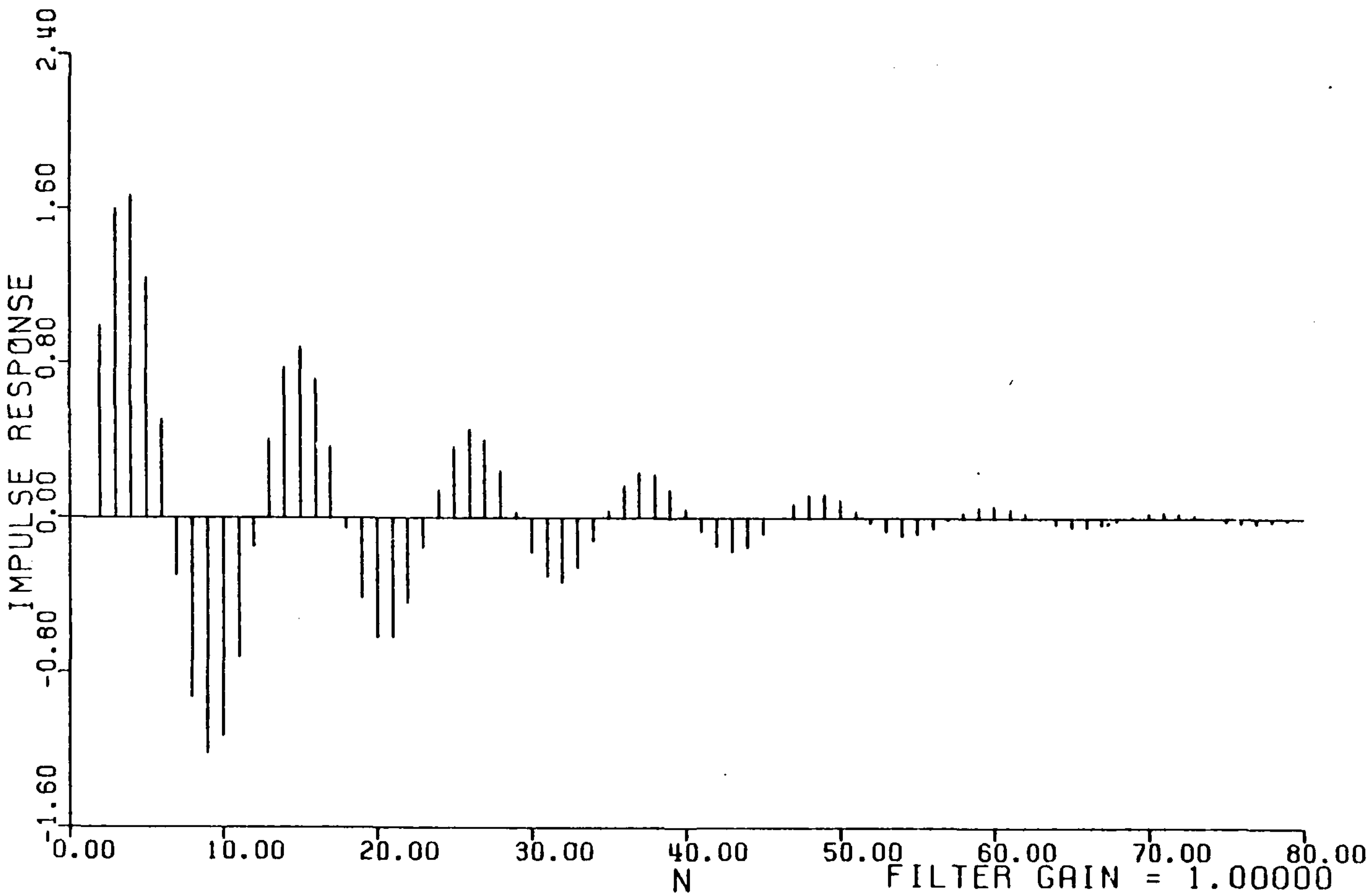


Figure 7-17b Unit Sample Response For A Complex Pole Pair At (0.8 $\pm j0.5$)

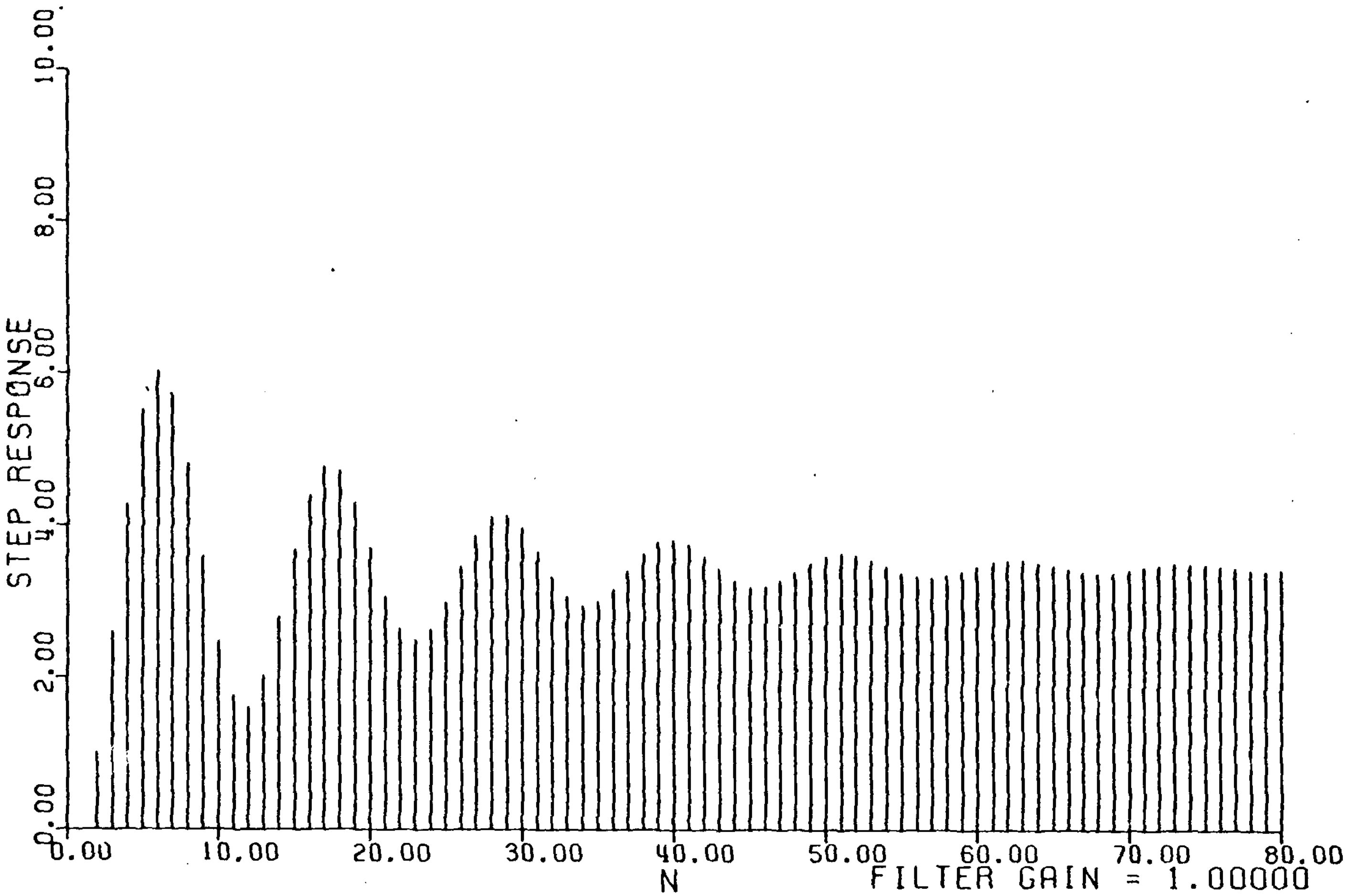


Figure 7-17c Unit Step Response For A Complex Pole Pair At (0.8 $\pm$j0.5)

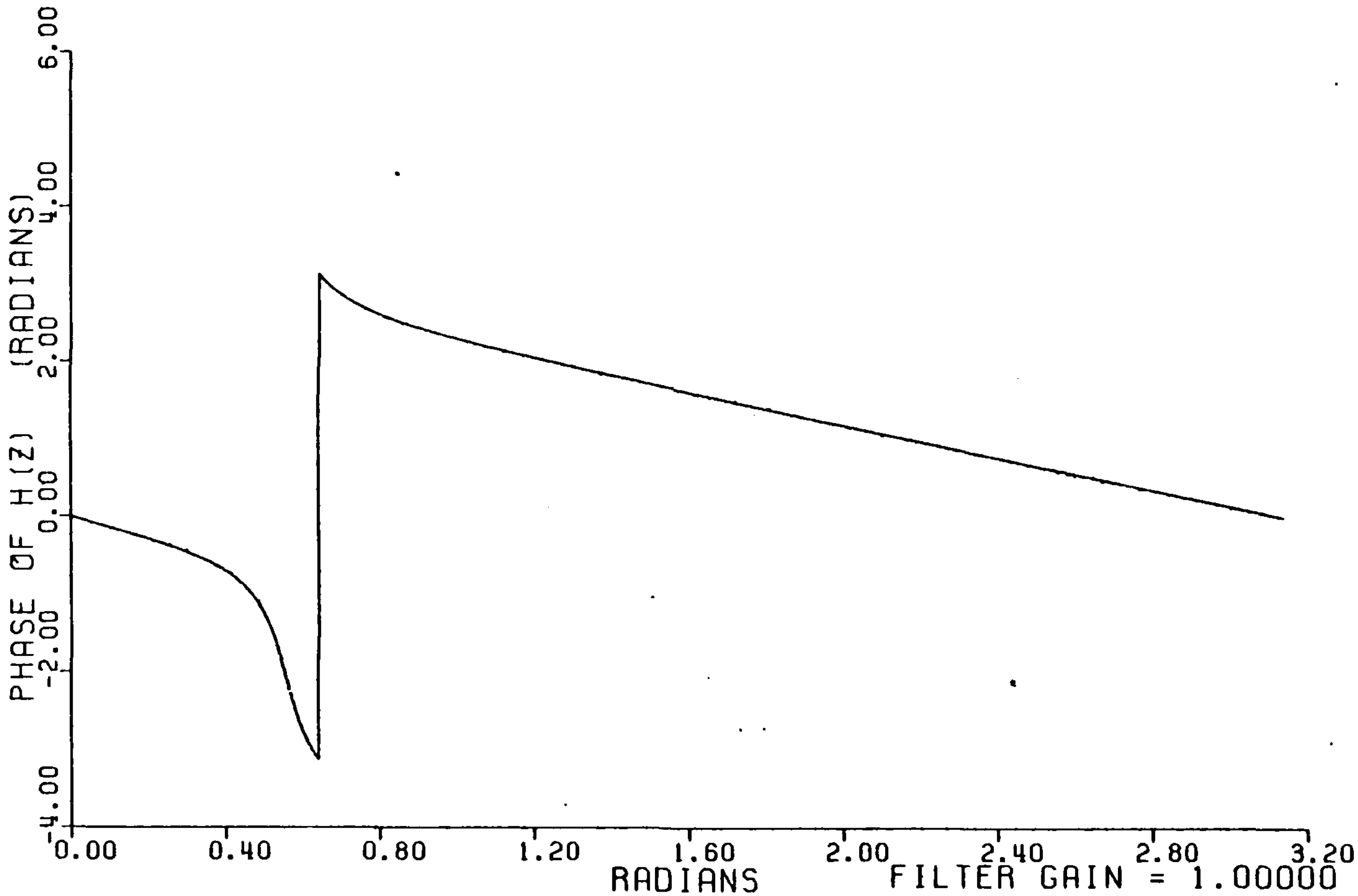


Figure 7-17d Phase Response For A Complex Pole Pair At (0.8 $\pm$j0.5)

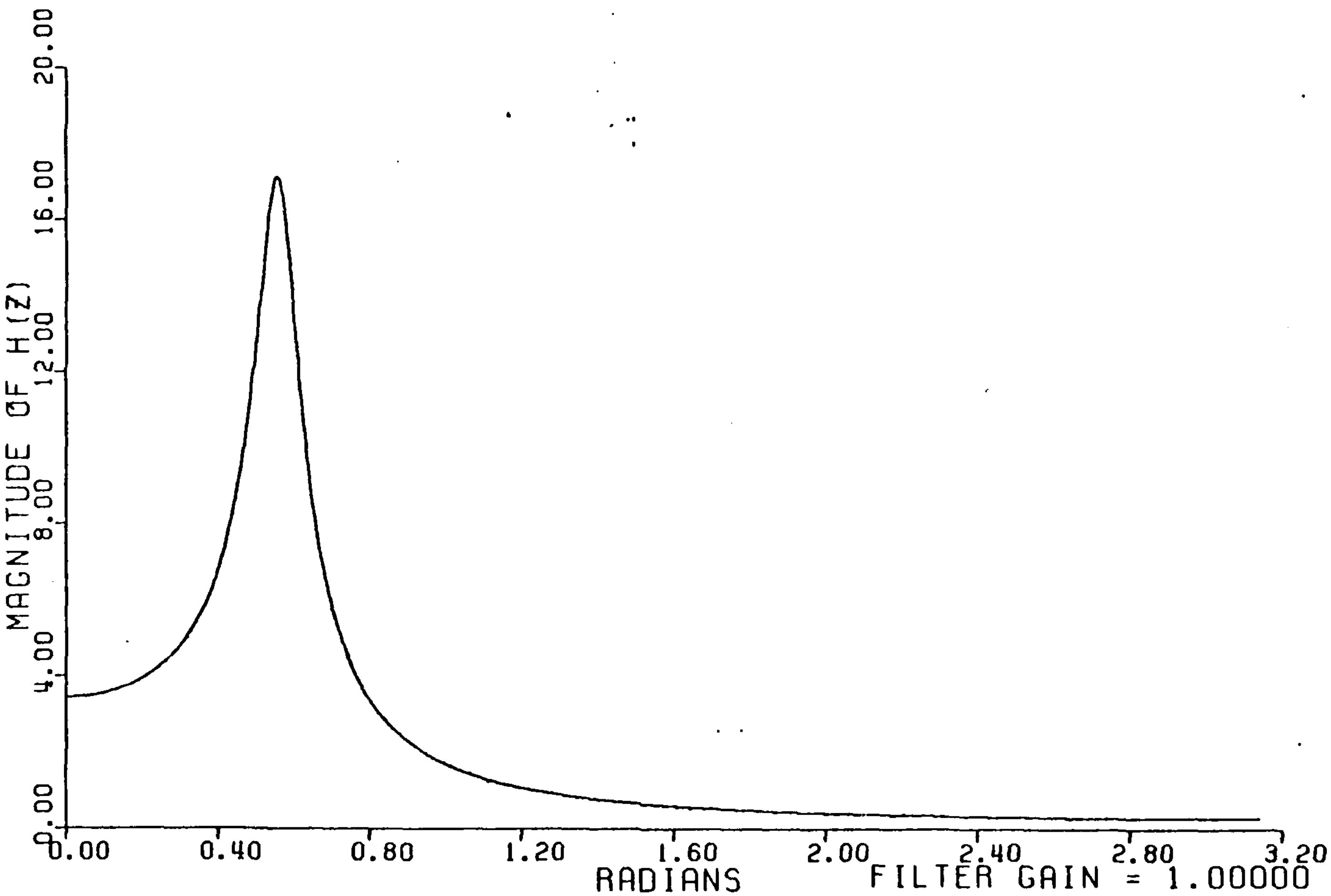


Figure 7-17e Magnitude Of H(z) For A Complex Pole Pair At (0.8 $\pm$j0.5)

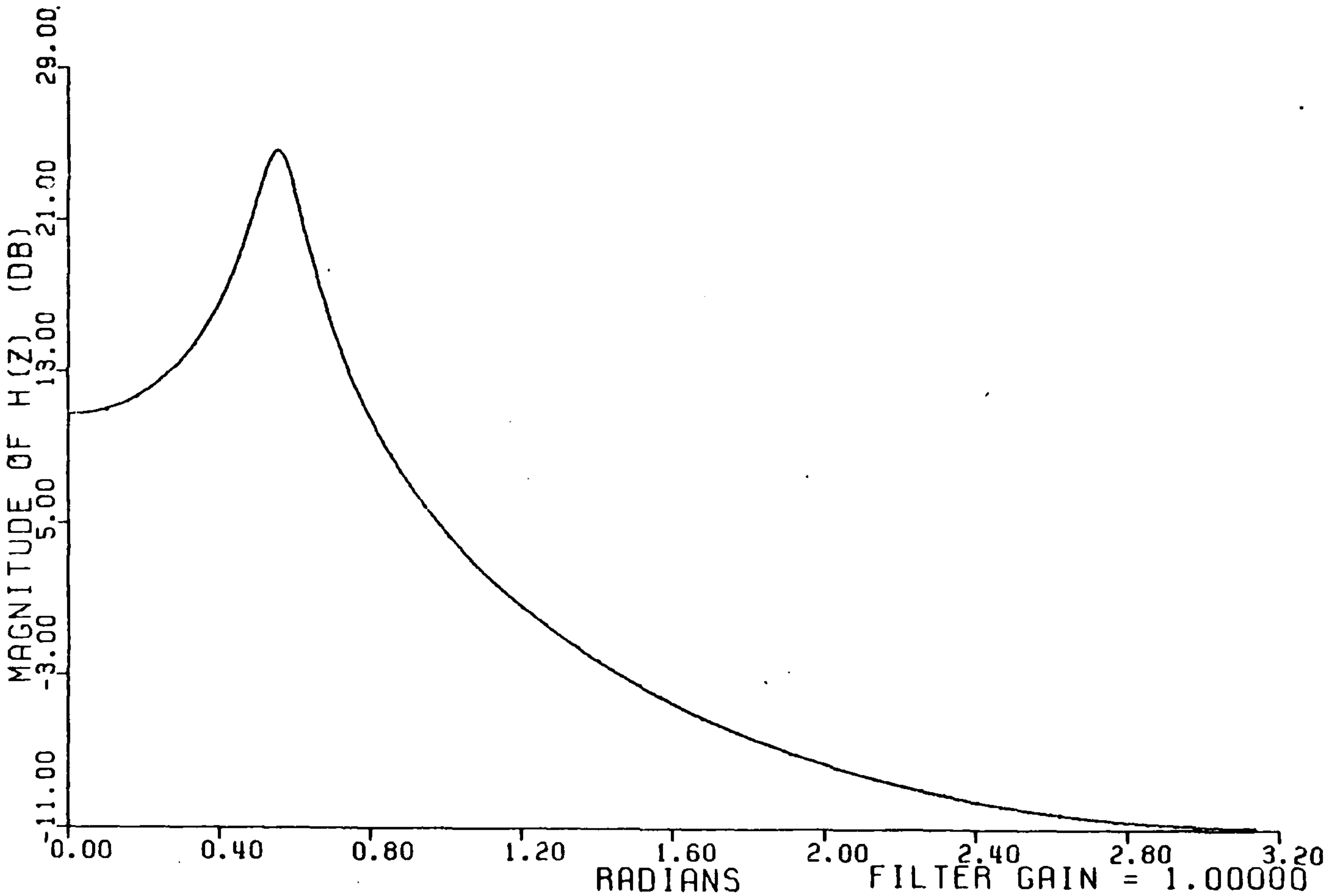


Figure 7-17f Magnitude Of H(z) In Decibels For A Complex Pole Pair At (0.8 $\pm$j0.5)

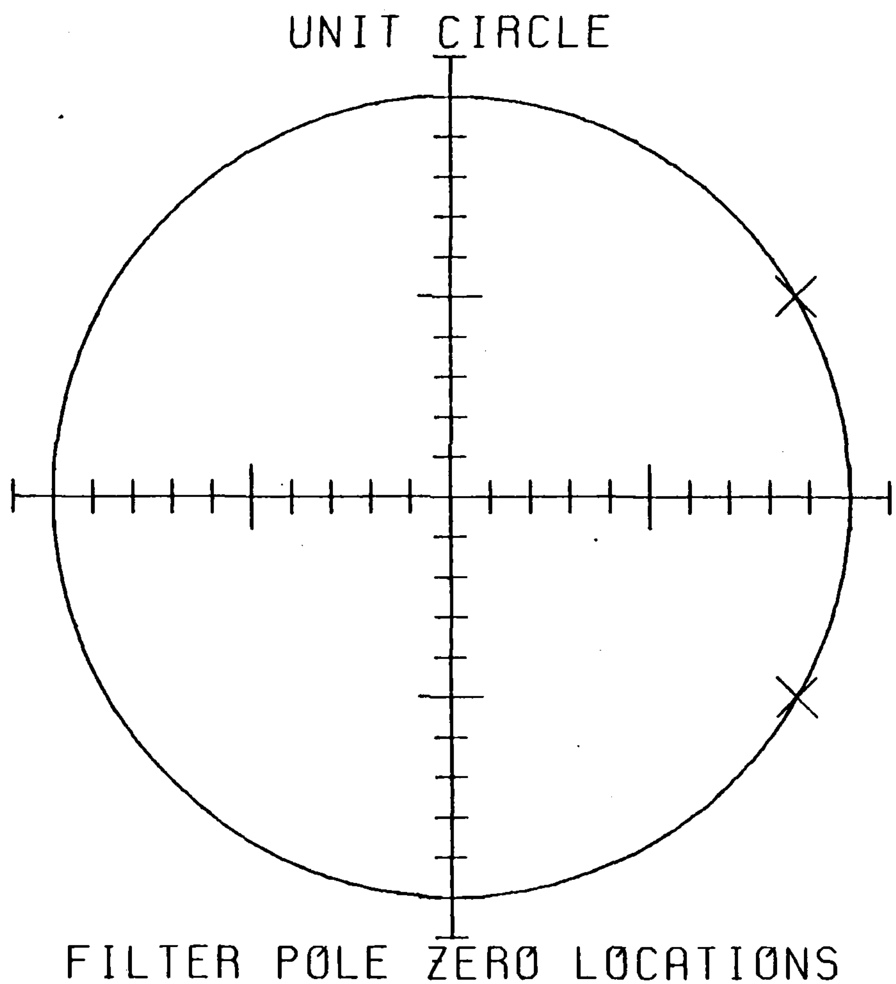


Figure 7.18a Example Of A Complex Pole Pair On The Unit Circle At $(0.866025 \pm j0.5)$

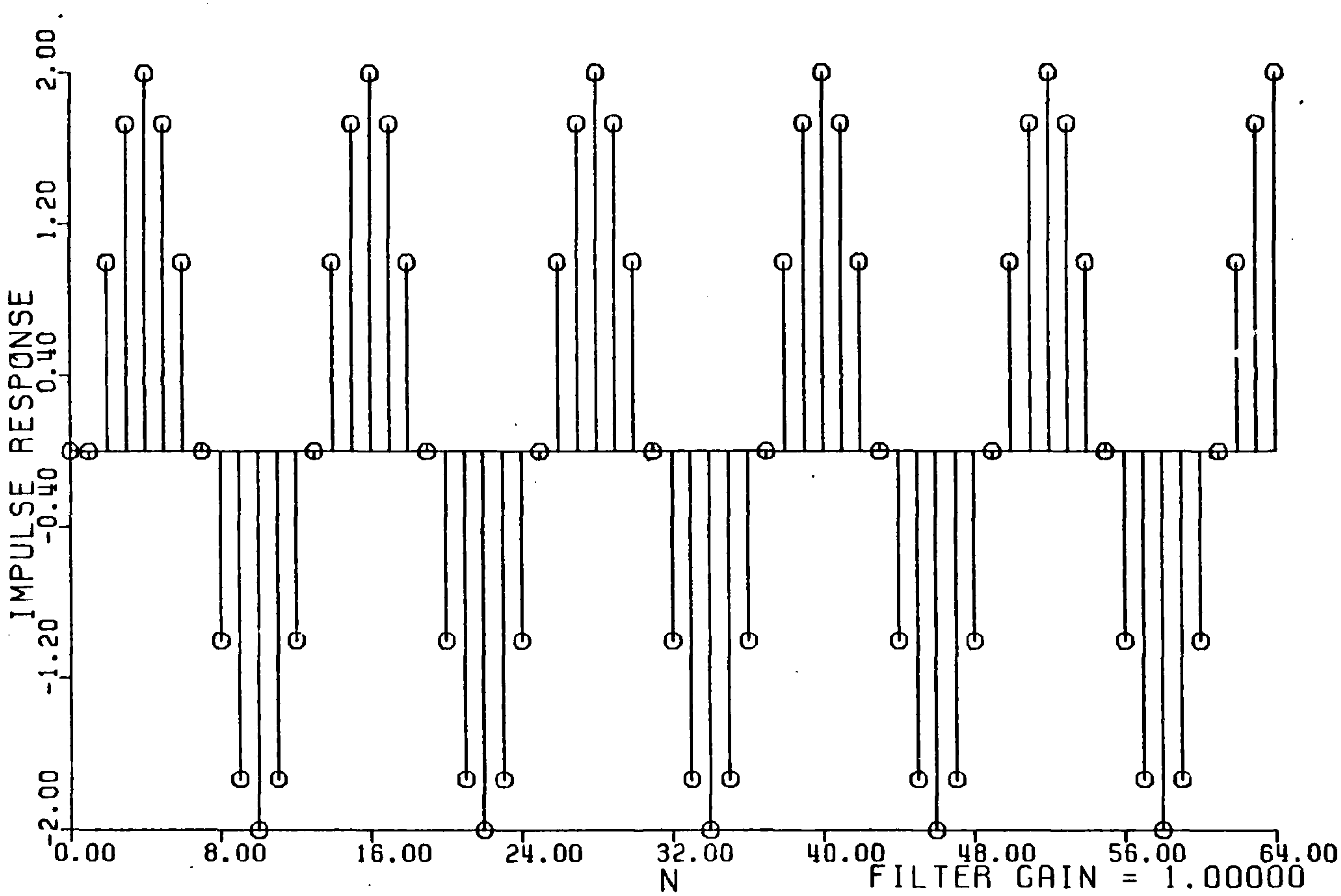


Figure 7-18b Unit Sample Response For A Complex Pole Pair At (0.866025 $\pm j0.5$)

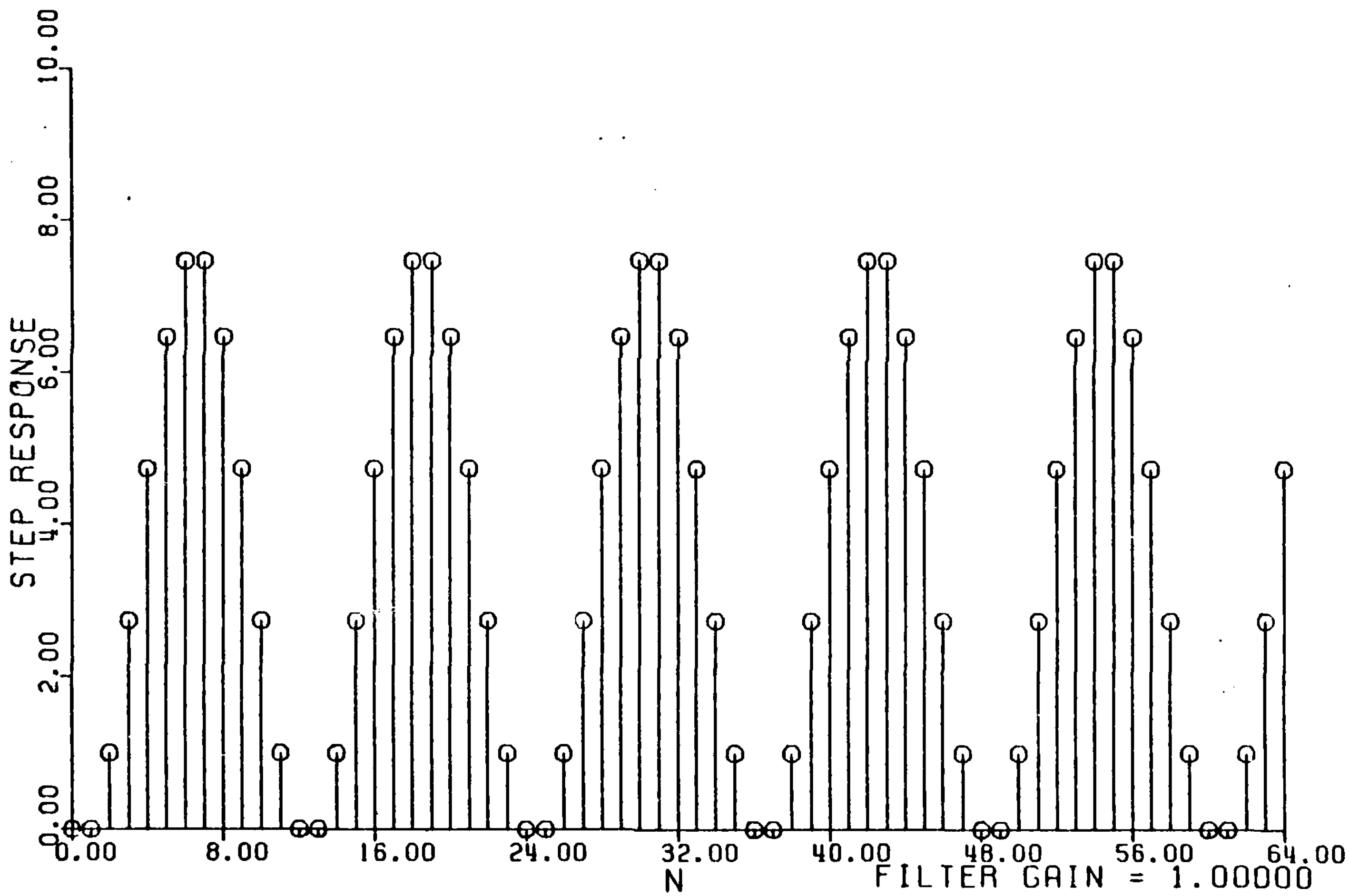


Figure 7-18c Unit Step Response For A Complex Pole Pair At (0.866025 $\pm j0.5$)

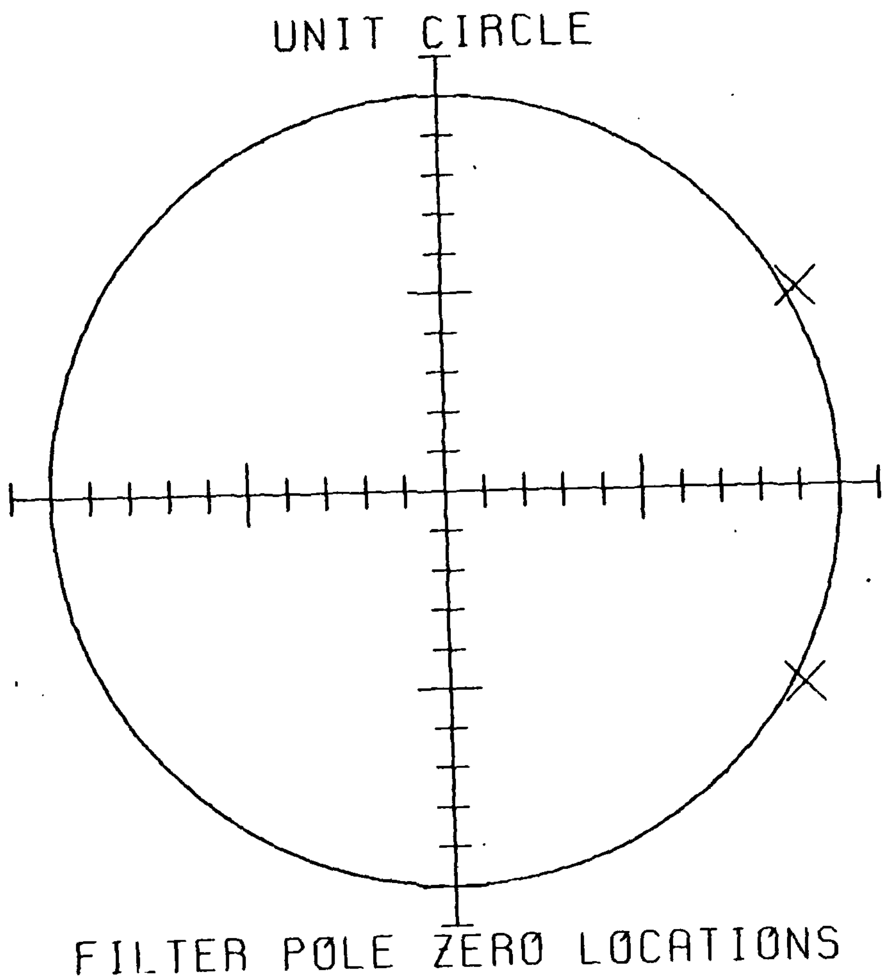


Figure 7-19a Example Of A Complex Pole Pair Outside The Unit Circle At (0.9 $\pm$j0.5)

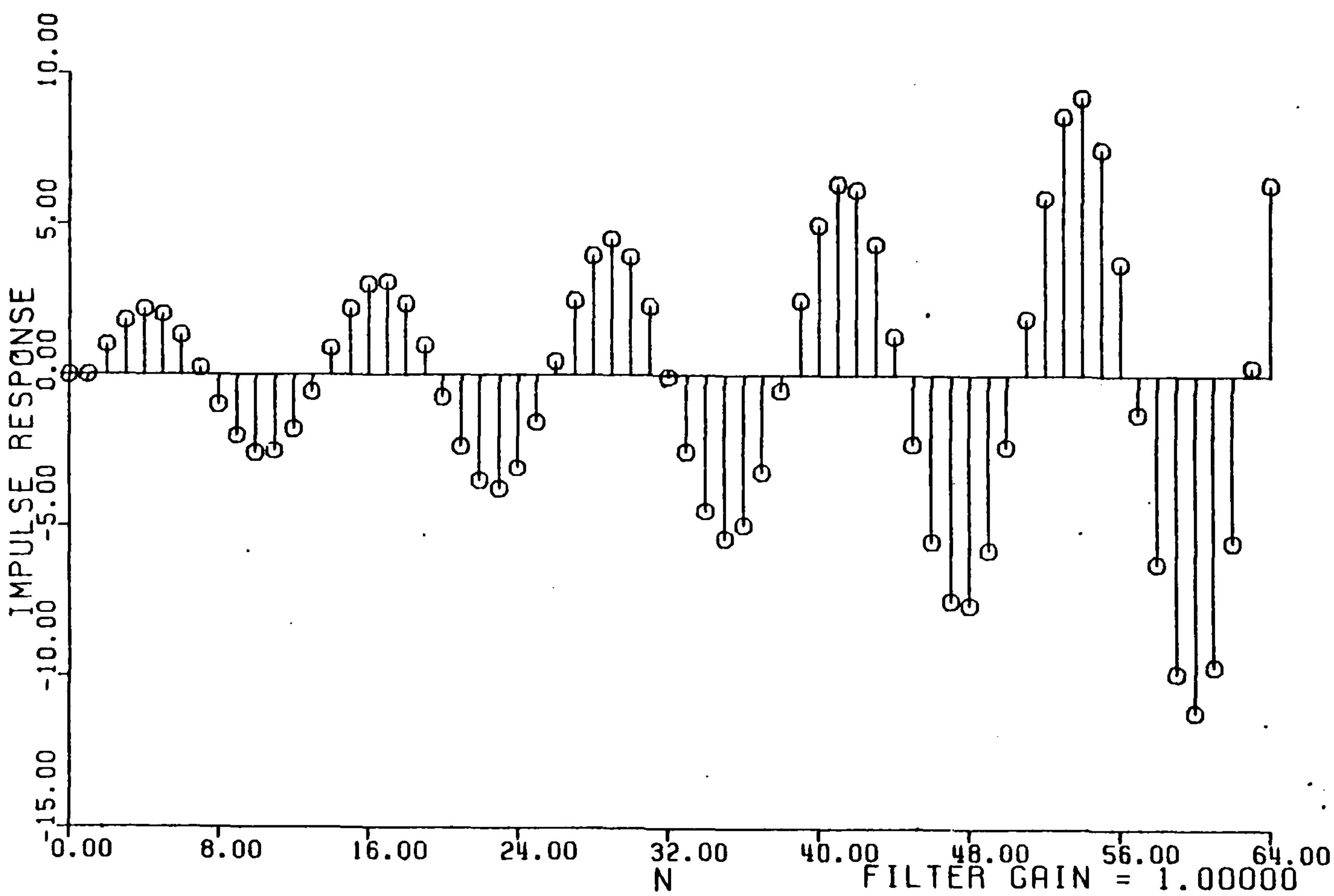


Figure 7-19b Unit Sample Response For A Complex Pole Pair At (0.9 ±j0.5)

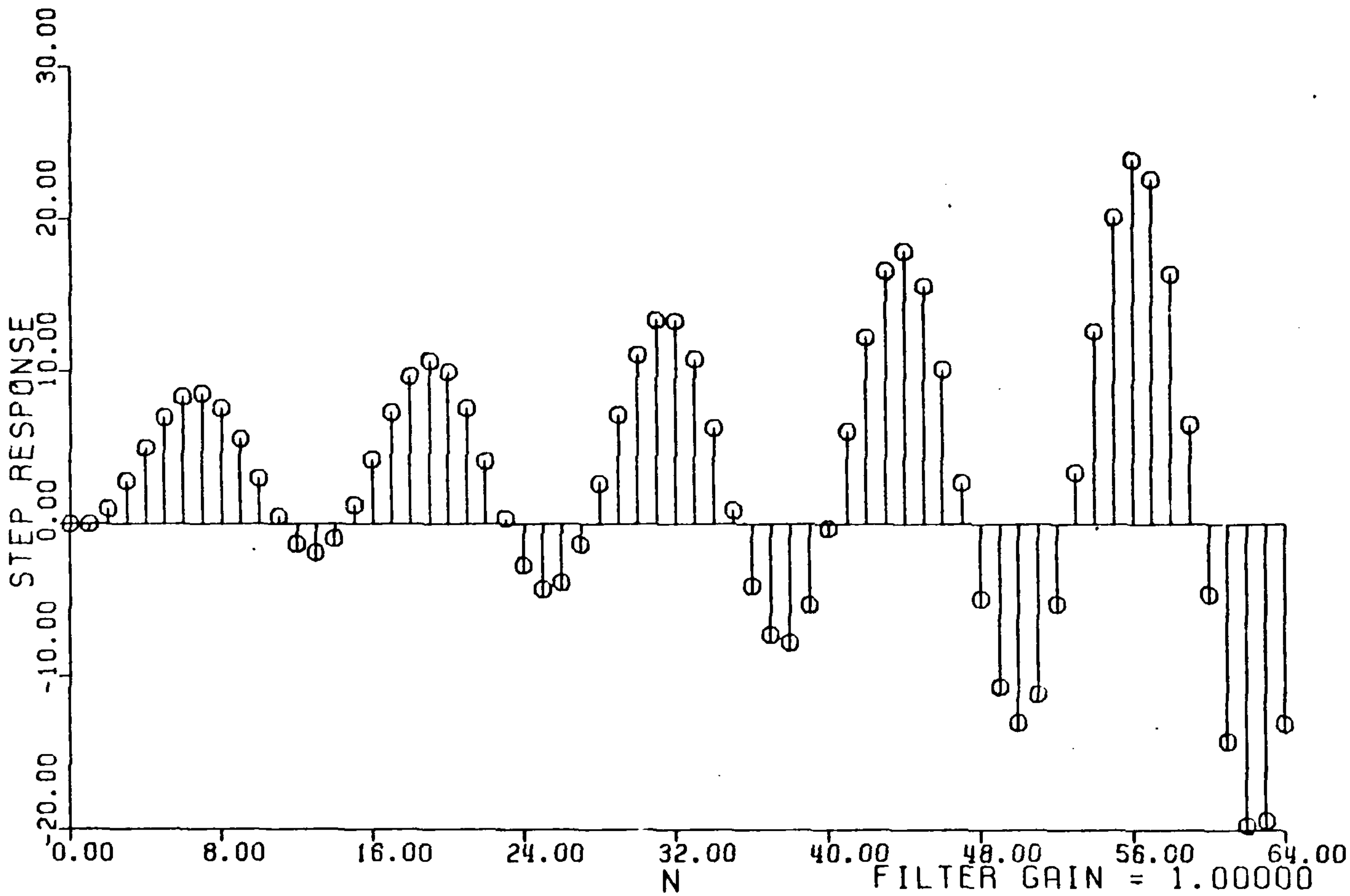


Figure 7-19c Unit Step Response For A Complex Pole Pair At (0.9 $\pm j0.5$)

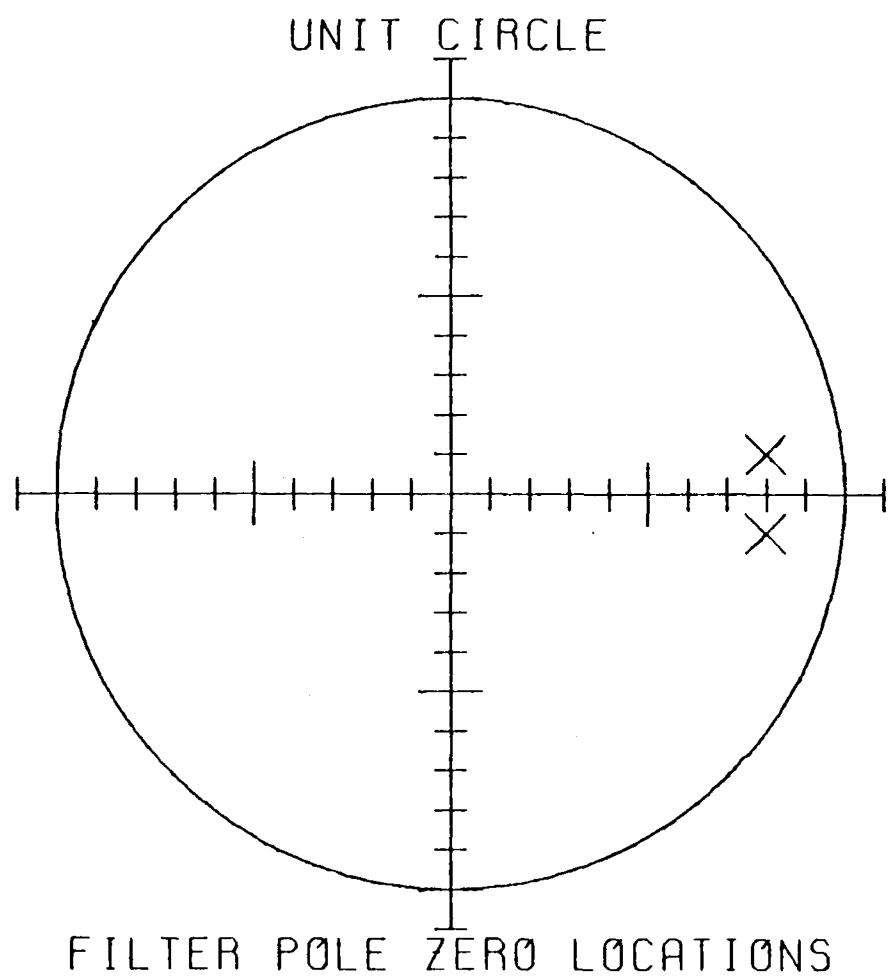


Figure 7-20a Example Of A Complex Pole Pair At $(0.8 \pm j0.1)$

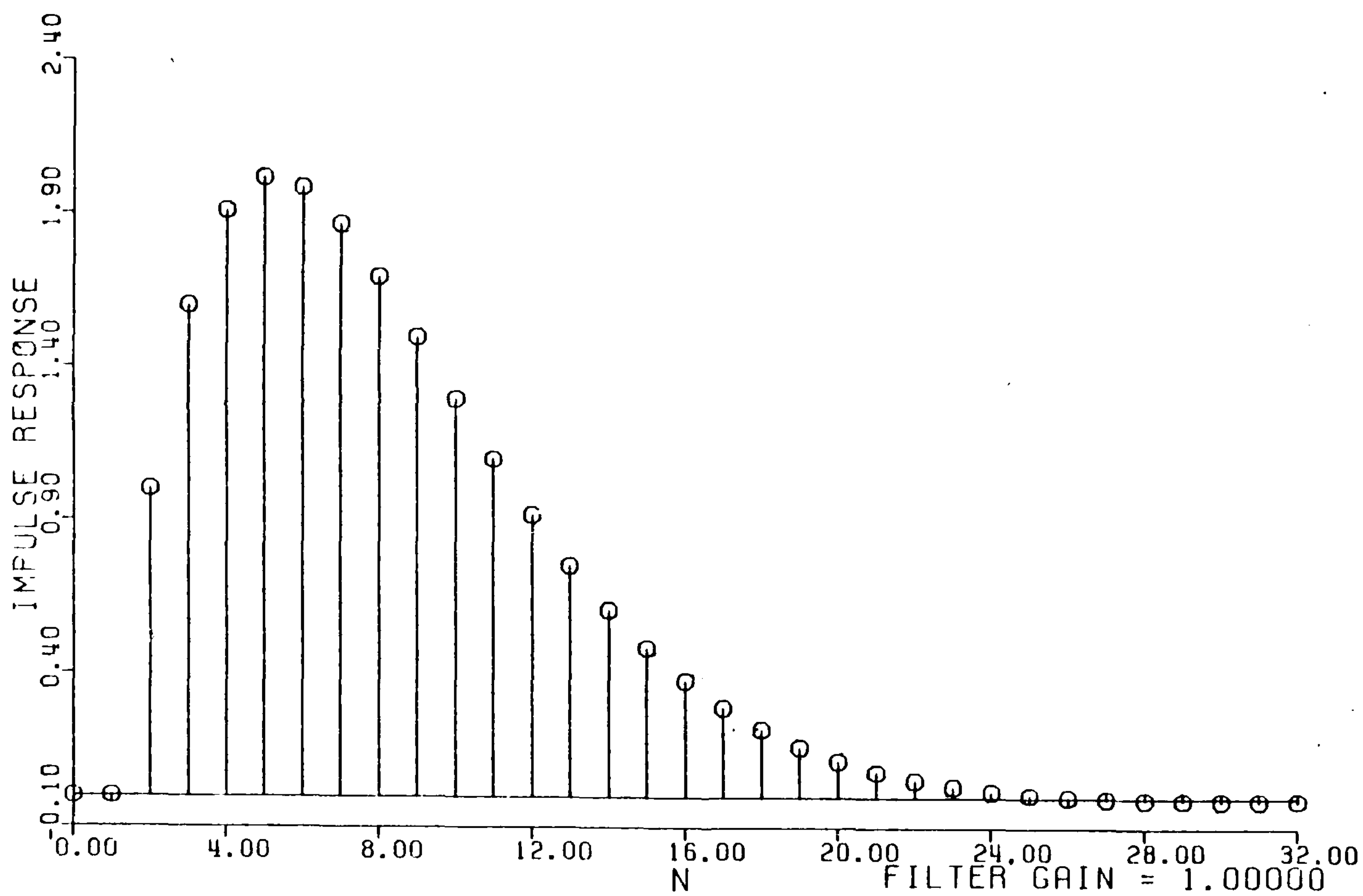


Figure 7-20b Unit Sample Response Of A Complex Pole Pair At (0.8 $\pm$j0.1)

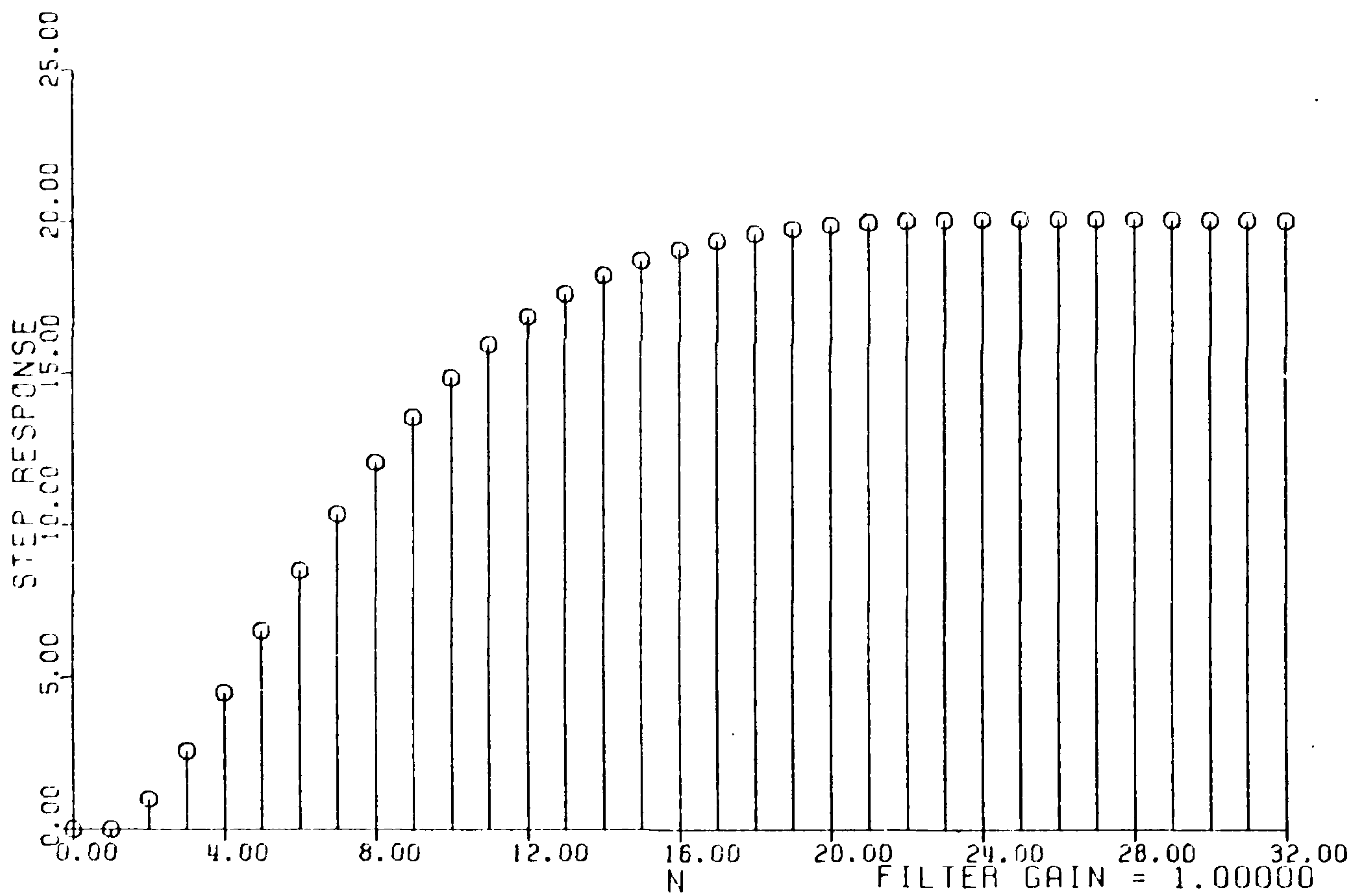


Figure 7-20c Unit Step Response For A Complex Pole Pair At (0.8 $\pm j0.1$)

AD-A093 252   NAVAL POSTGRADUATE SCHOOL MONTEREY CA   F/G 9/3

ADVANCED SIMULTION OF DIGITAL FILTERS.(U)

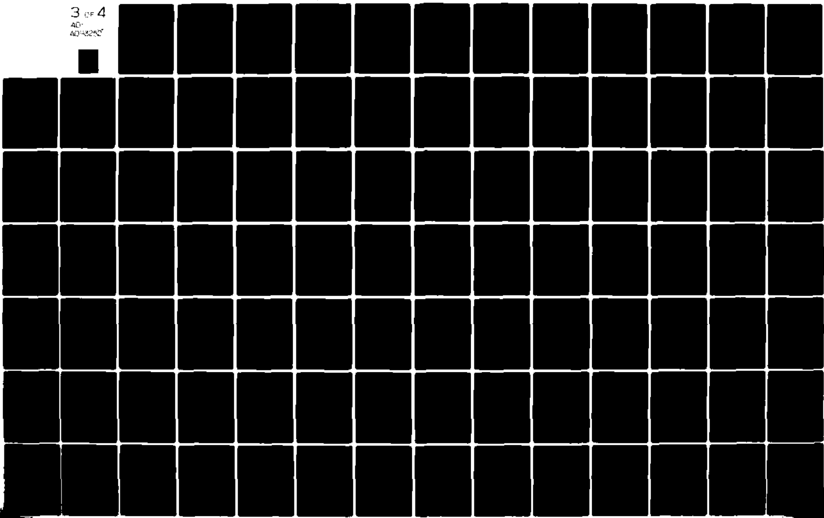
SEP 80 G S DOYLE

UNCLASSIFIED   NL

3 OF 4

AD

A093252



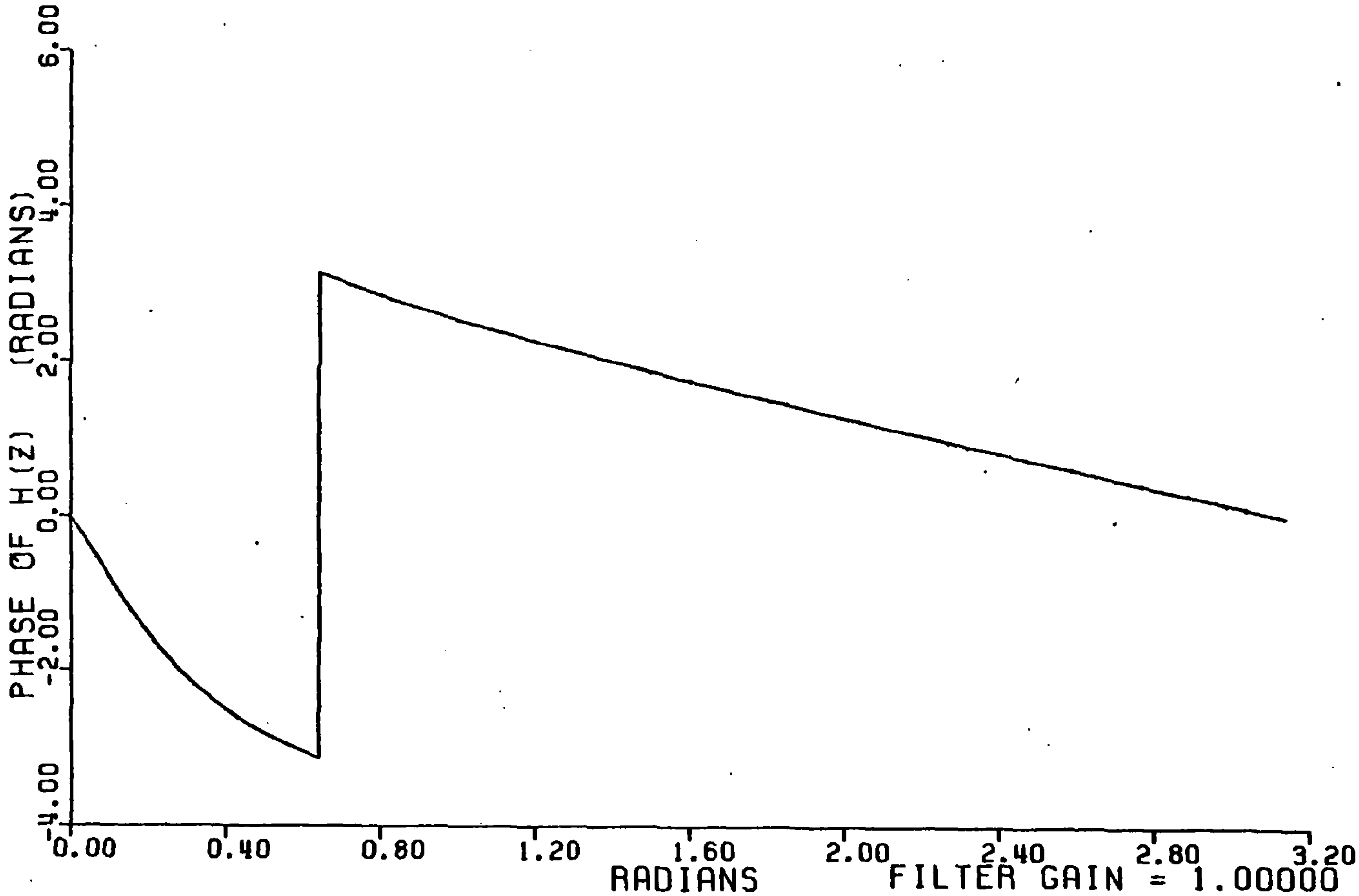


Figure 7-20d Phase Response For A Complex Pole Pair At (0.8 ±j0.1)

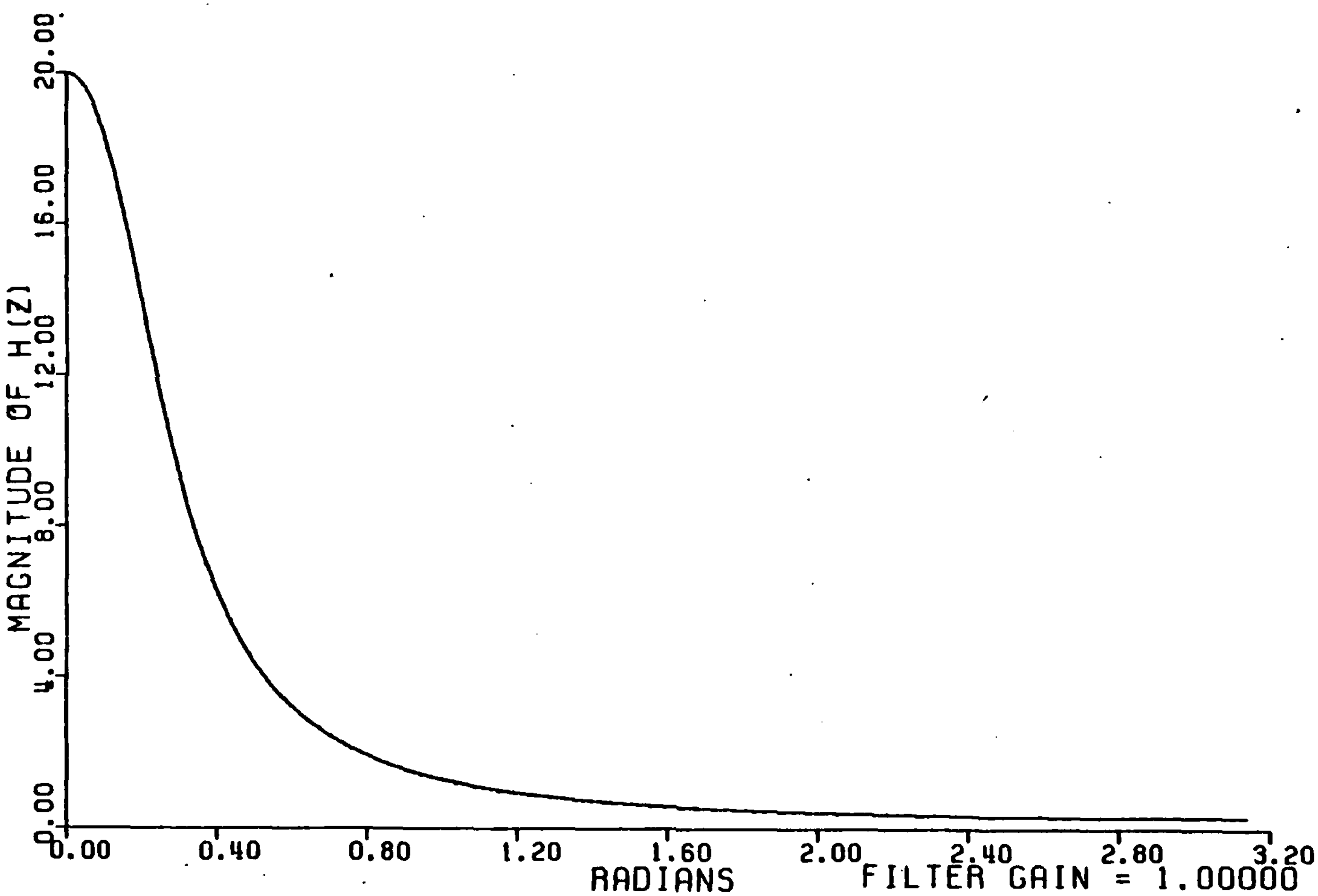


Figure. 7-20e Magnitude Of H(z) For A Complex Pole Pair At (0.8 $\pm j0.1$)

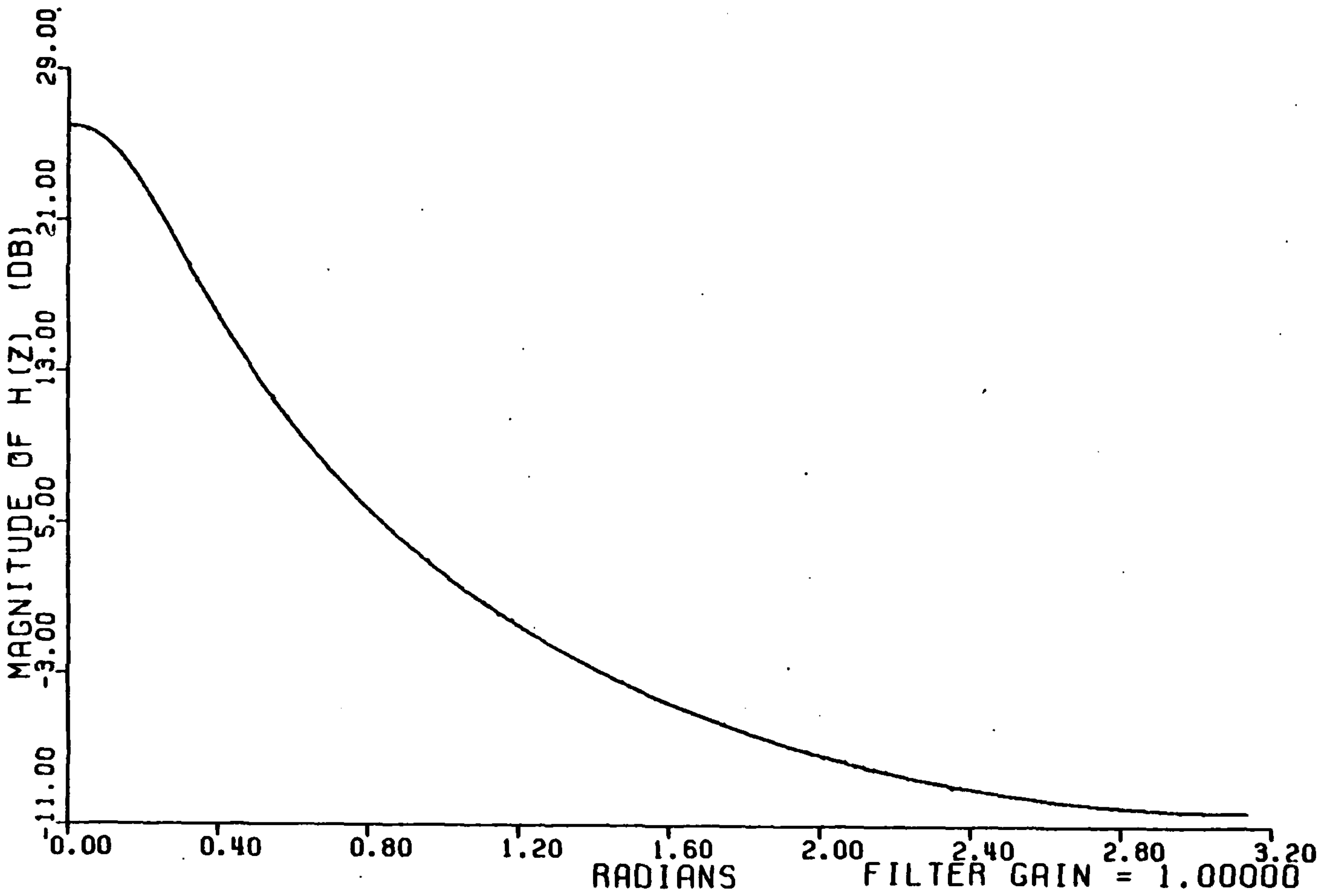


Figure 7-20f Magnitude Of H(z) In Decibels For A Complex Pole Pair At (0.8 $\pm j0.1$)

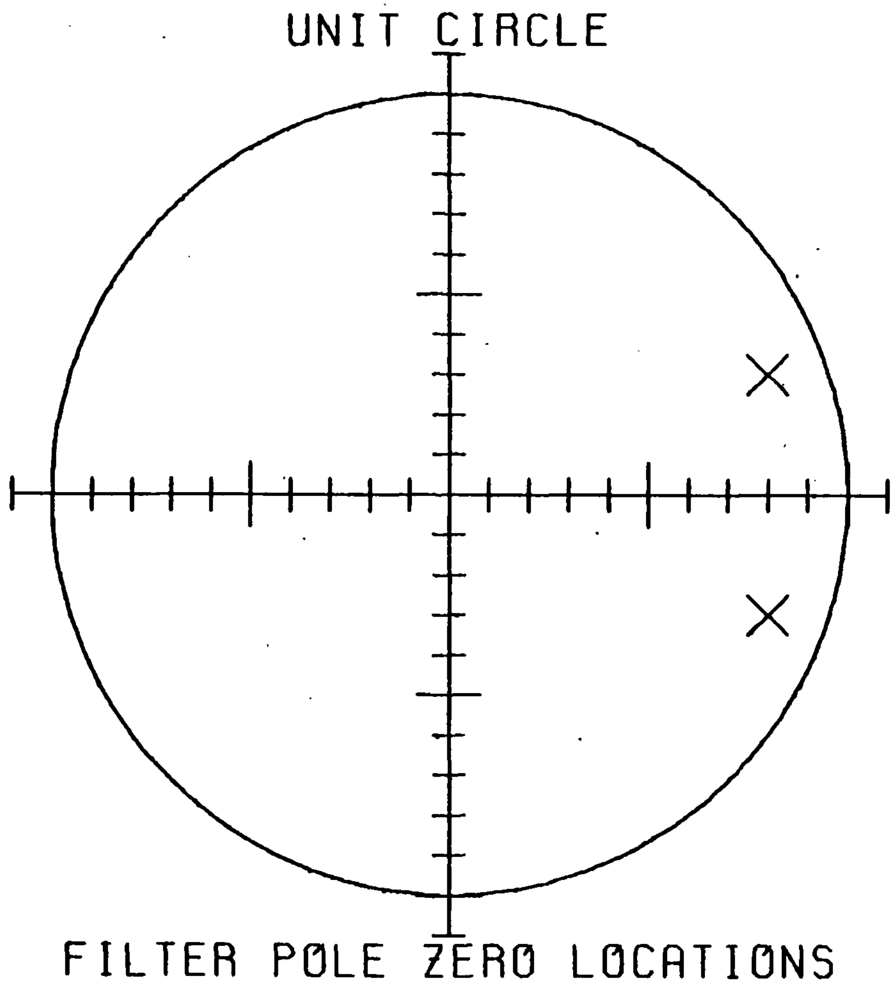


Figure 7-21a Example Of A Complex Pole Pair At (0.8 $\pm$j0.3)

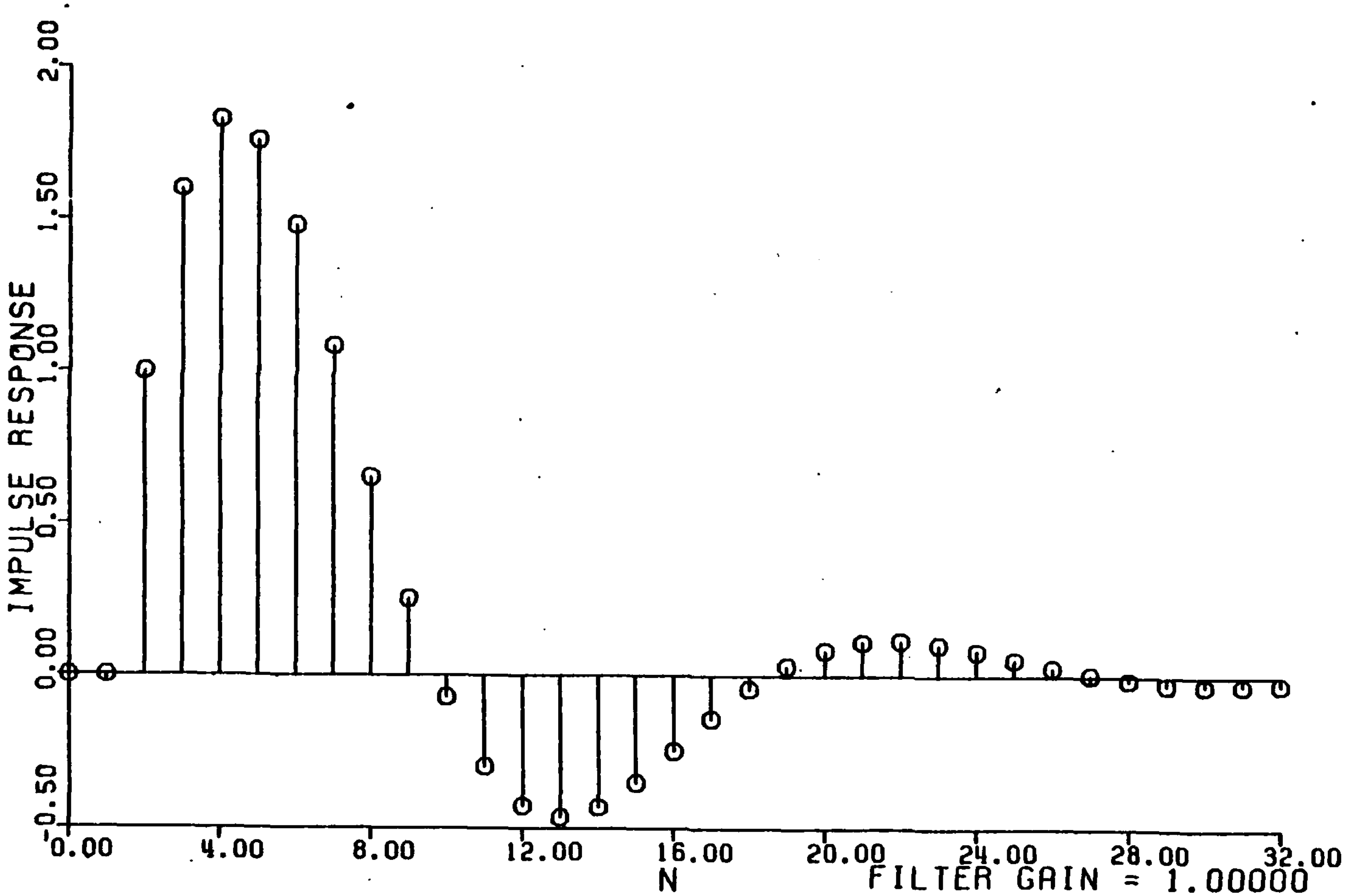


Figure 7-21b Unit Sample Response For A Complex Pole Pair At (0.8 $\pm$j0.3)

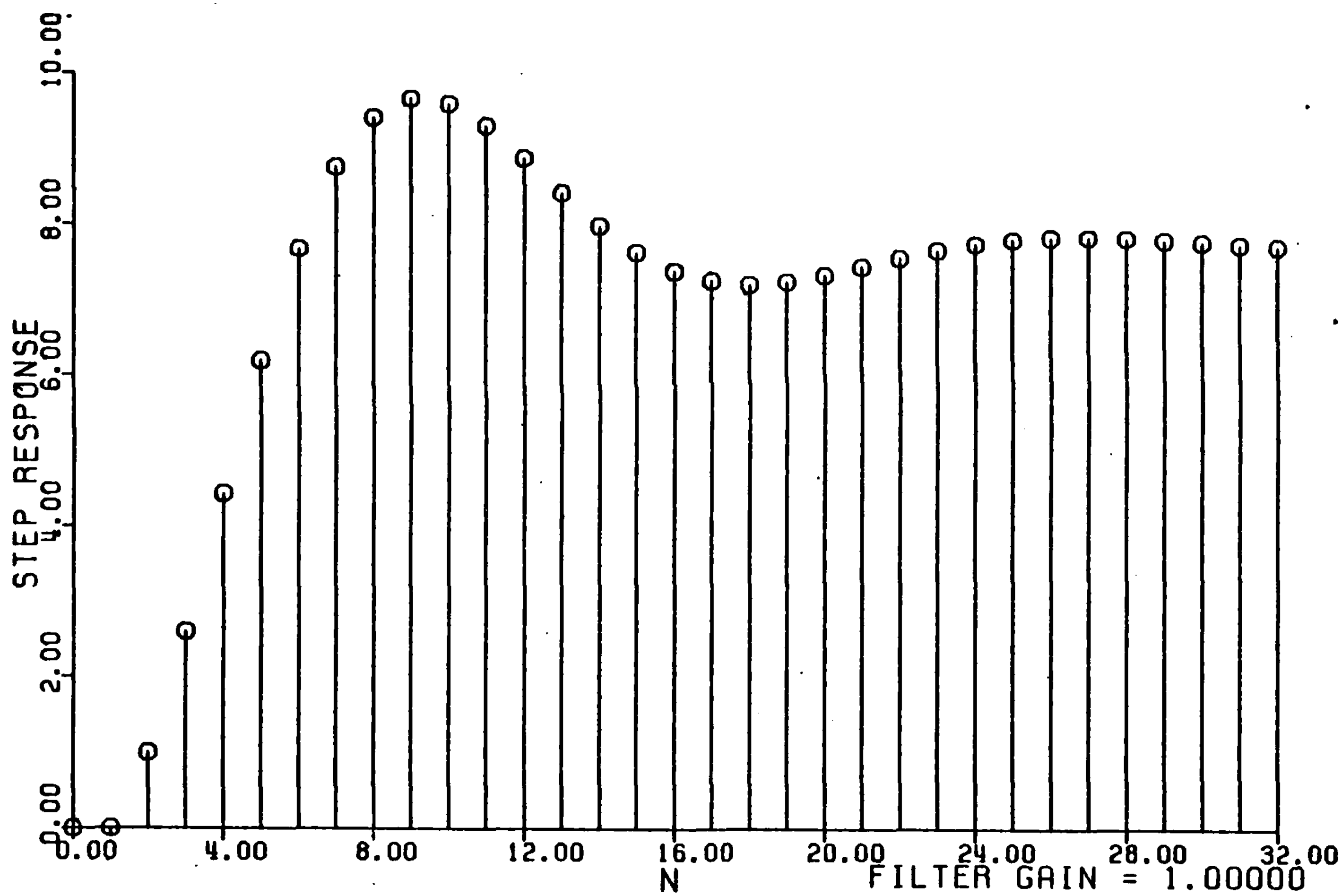


Figure 7-21c Unit Step Response For A Complex Pole Pair At (0.8 $\pm j0.3$)

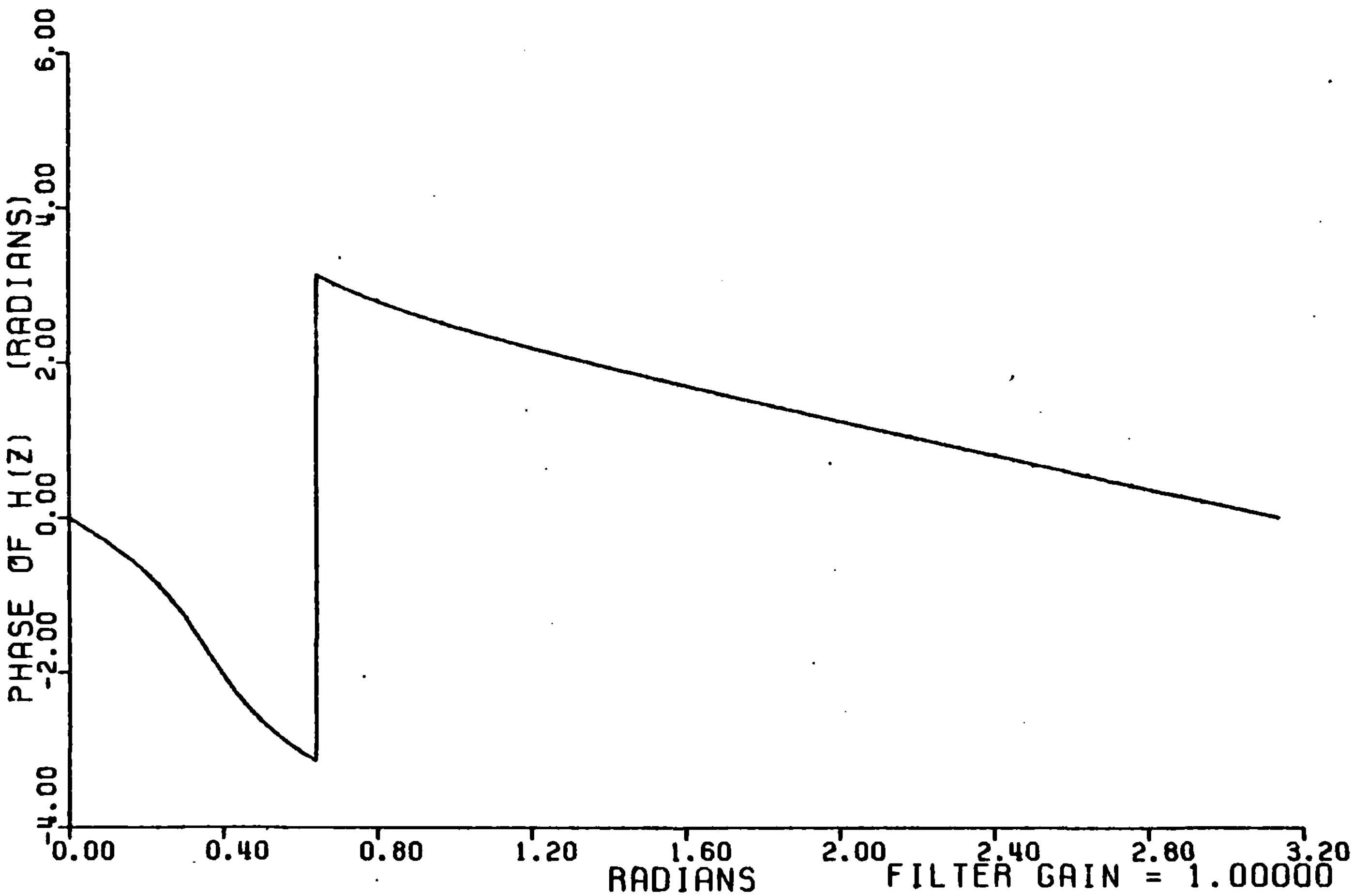


Figure 7-21d Phase Response For A Complex Pole Pair At (0.8 $\pm j0.3$)

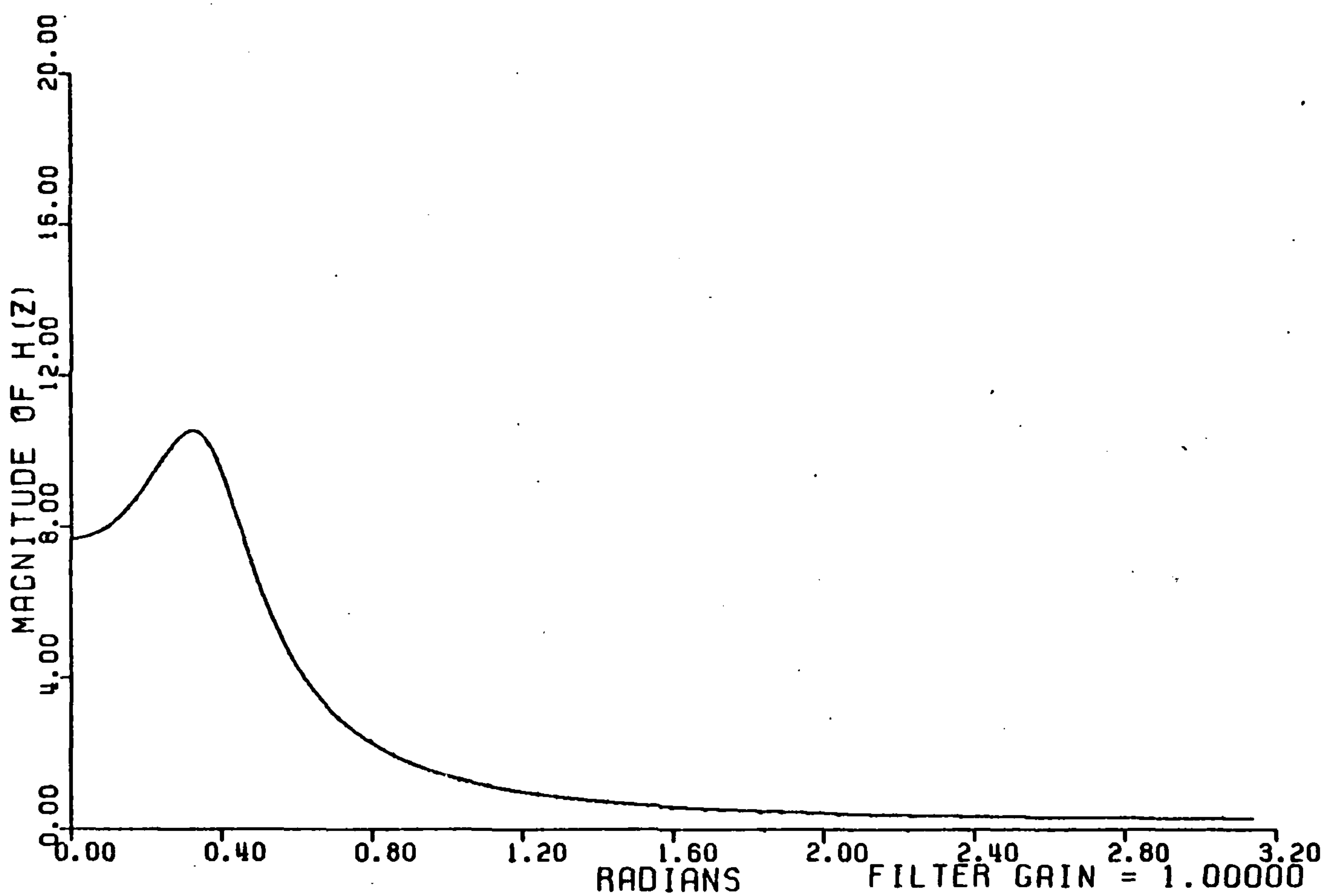


Figure 7-21e Magnitude Of H(z) For A Complex Pole Pair At (0.8 $\pm$j0.3)

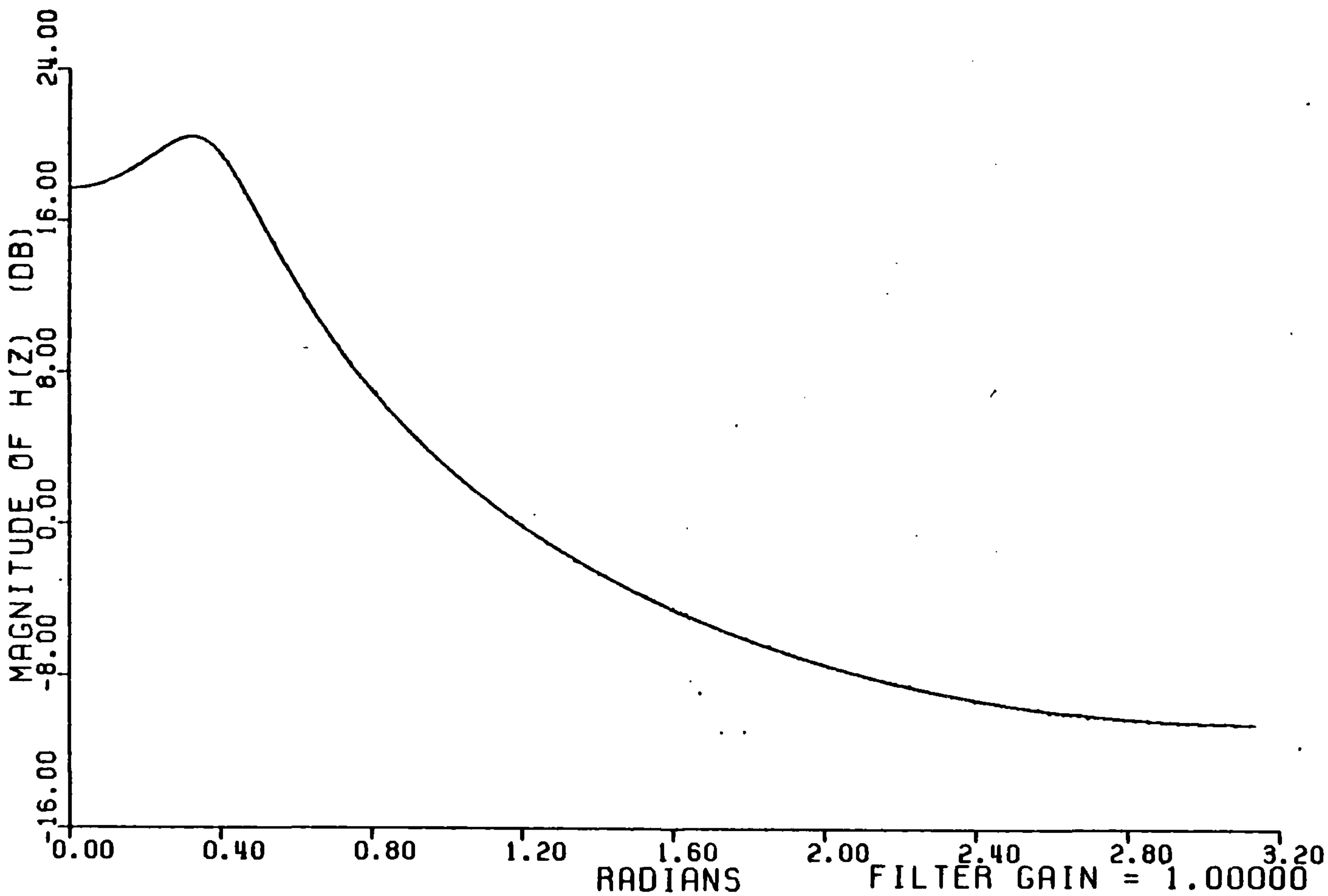


Figure 7-21f Magnitude Of H(z) In Decibels For a Complex Pole Pair At (0.8 $\pm$j0.3)

VIII. INTERPRETATION OF ASDF OUTPUT

When the user initially confronts the ASDF program it is expected that he will attempt to duplicate some of the examples presented earlier in this thesis or similar examples found in the literature. These types of applications of ASDF are straightforward and the results generated can be compared with published results. Interpretation of the output plots is seldom confusing.

As the user becomes more familiar with the use of the ASDF program and the positioning of the poles and zeros in the z-plane he will undoubtedly attempt to generate filters with sharper transition regions, and greater attenuation in the stop band(s). In general these filter improvements are accomplished by increasing the order of the system and positioning the poles closer to the unit circle. As the order of the system increases, and as the poles of the filter become closer to the unit circle, the filter becomes increasingly subject to the difficulties associated with the finite precision arithmetic used in the filter implementation.

A tacit assumption made very early in most digital signal processing courses (and often quickly forgotten) is that infinite precision arithmetic is available in performing the calculations. This assumption breaks down

when digital filters are realized in hardware or simulated! In general there are three sources of error which may affect the performance of digital filters. Due to the finite precision representation of the coefficients, the values of the implemented filter coefficients are only approximately equal to the desired values. Second, errors may be generated in the arithmetic operations. In general, the product of two n-bit numbers results in a 2n-bit result. An error occurs when this 2n-bit number is converted to n-bits for succeeding computations. The third type of error encountered is the analog to digital conversion required for processing analog signals with digital filters. Only the first two of these errors is of concern in the ASDF program.

While an in depth analysis of finite precision effects is beyond the scope of this work, let us consider the filter which has eight zeros located at the origin, and eight poles located at -0.9. Entering these locations into the z-plane using the ASDF program generates the filter responses in figures 6-1a, b, c. A superficial glance at these responses indicates that this filter is unstable. The filter must be stable, however, since all of the poles lie within the unit circle.

There are several phenomena involved in this example. The first is the dynamic range limitation of the system on which the filter is implemented. The IBM-360 is limited to computing with numbers between 16^{64} and -16^{-63}. Once the

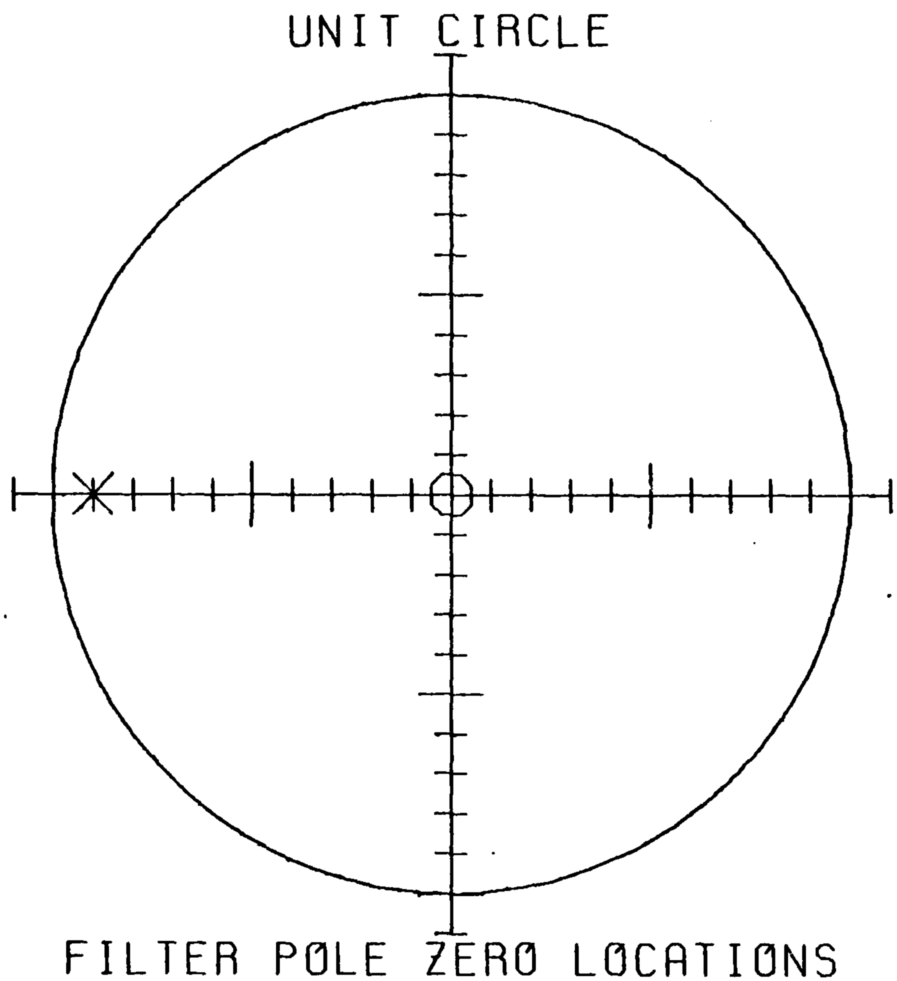


Figure 8-1a Example Filter With Eight Poles At -0.9 And Eight Zeros At The Origin

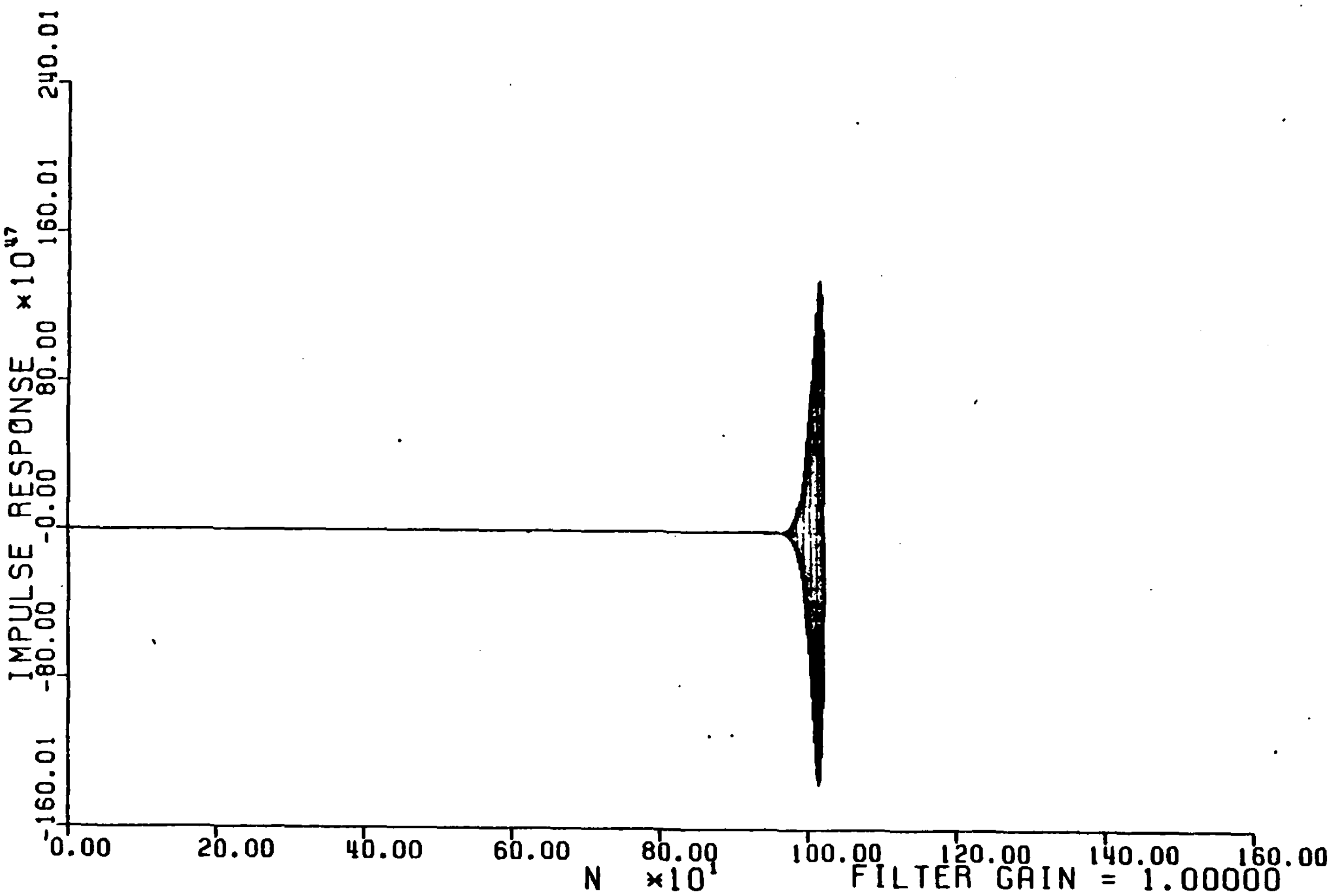


Figure 8-1b Unit Sample Response Of A Filter Exceeding System Dynamic Range Limitations

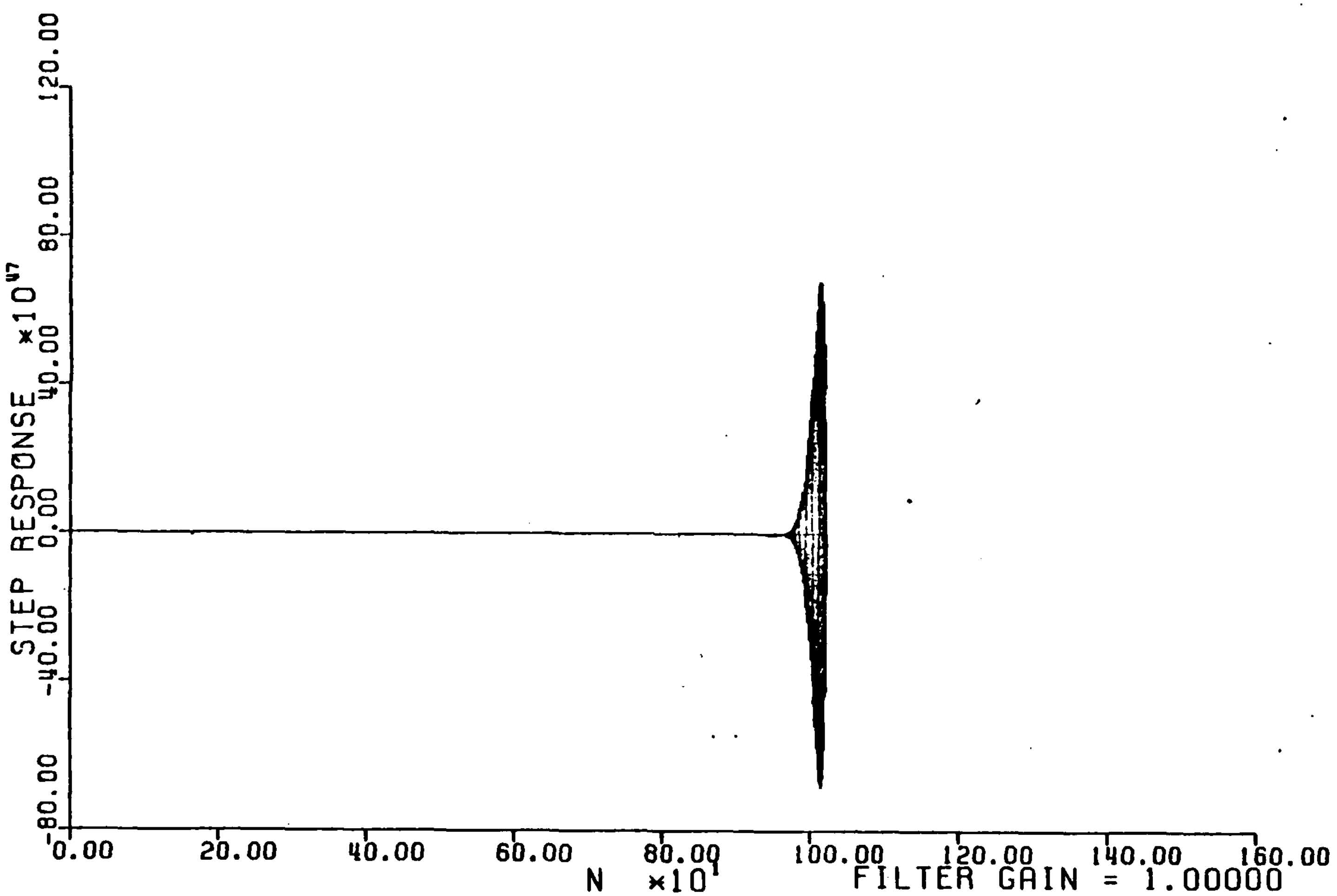


Figure 8-1c Step Response For A Filter Exceeding System Dynamic Range Limitations

larger of these limits is exceeded, the computations are terminated and it is not clear whether the designed filter is actually unstable or has merely exceeded the dynamic range limitations of the machine. The dedicated user may decide to modify the programs presented in the appendicies to investigate the time responses of such systems. Clearly the form of the time responses will not be significantly altered if a modified unit sample response

$$1 \times 10^{-10}, 0, 0, 0, \ldots$$

and a modified step response

$$1 \times 10^{-10}, 1 \times 10^{-10}, 1 \times 10^{-10}, \ldots$$

is used.

If these modifications are performed, two other phenomena become apparent, each relating to the finite precision implementation of the digital filter. The first is that, in general, digital filters have deadbands, the second that the filter output may degenerate to a limit cycle. The original works by Blackman [Ref. 1] . demonstrate that digital filters exhibit both of these effects. Extensive covgerage of these phenomena is not appropriate here, however, a dead-band in a digital filter can be simplistically described as a span of filter input values for which the filter output does not change. The limit cycle effect is evident when a digital filter responds to a steady state input. For constant inputs, the filter output may run through a cycle of values. The propensity for

experiencing limit cycles is obvious in the calculation of the step response. A close look at the unit sample response shows that after a sufficiently large number of input values, this unit sample sequence becomes in reaching steady state, a constant.

When observing the output response of the example filter, there is still another possibility for explaining this output. This explanation revolves around the fact that the poles in the implemented filter are not at the locations expected. In fact, the poles may actually move outside the unit circle even though they were entered within. In general, there are only a finite number of pole locations possible with finite word length coefficient representations. The effect of this limitation is best explained using another example. Consider a filter with three zeros located at 0.5, and 0.2 +/- J0.5, and four poles located at -0.999, -0.6, and -0.69 +/- J0.69. The infinite precision transfer function for this filter is expressed in equation 8.1.

$$\frac{Z^3 - .9\ Z^2 + .49\ Z - .145}{Z^4 + 2.979\ Z^3 + 3.75822\ Z^2 + 2.3497398\ Z + .57074868} \qquad (8.1)$$

Now consider the finite precision case where each of the coefficients must be represented by a six bit number. The pole at -0.6 is expressed in binary (exclusive of sign) as

.10011001100. . . and is truncated to six bits as .100110. When all of the coefficients are so represented, the transfer function is expressed by equation 8.2. Clearly the poles and zeros of this second transfer function are not the

$$H(z) = \frac{Z^3 - .890625\ Z^2 + .474275\ Z - .140625}{Z^4 + 2.9375\ Z^3 + 3.75\ Z^2 + 2.3125\ Z + .5625} \quad (8.2)$$

same in equation 8.1. In general the differences will change all of the filter responses. In fact for this particular case the new poles of the system are at -0.70522 +/- J0.16677, and -0.76353 +/- J0.698616; this filter is unstable!

The reason for the shift in pole locations is clearly portrayed by the filter whose characteristic equation is equation 8.3.

$$(Z + .995)^8 = 0 \quad (8.3)$$

This filter has eight poles at -0.995. Note what happens to the filter when an error is introduced by truncating the infinite precision coefficient values to thirty-two bits. For simplicity restrict the discussion to the ramifications of an error generated in the value of the constant $(.995)^8$. This error can be as large as 2×10^{-10}. With this error explicitly included, the new characteristic equation will be:

$$(Z + .995)^8 + 2 \times 10^{-10} = 0 \quad (8.4)$$

The roots of equation (8.4) are complex, and lie on a circle of radius 0.06 around the point -0.995 +/- J 0.0. Figure 8-2 shows that three of these roots lie outside the unit circle. Thus with a finite precision implementation of the filter (even with 32 bits of precision), it is possible for the pole locations to move so far that the filter actually becomes unstable.

The results of this section as well as four types of ASDF outputs which have been observed, are summarized in figure 8-3. Trace A shows the unit sample response of a stable filter. Trace B shows the response of a filter which exceeds the dynamic range limitations of the implementation system. Trace B1 is for an unstable filter and B2 is for a stable filter caught in a limit cycle. These two traces can be observed or explored only by modifying the unit sample sequence. Trace C shows a stable filter which falls into a limit cycle below the dynamic range limit of the machine upon which the implementation has been performed. Note that the performance of a filter above the dynamic range limitation of the IBM 360 (or any other machine) cannot be observed, unless the ASDF program is modified.

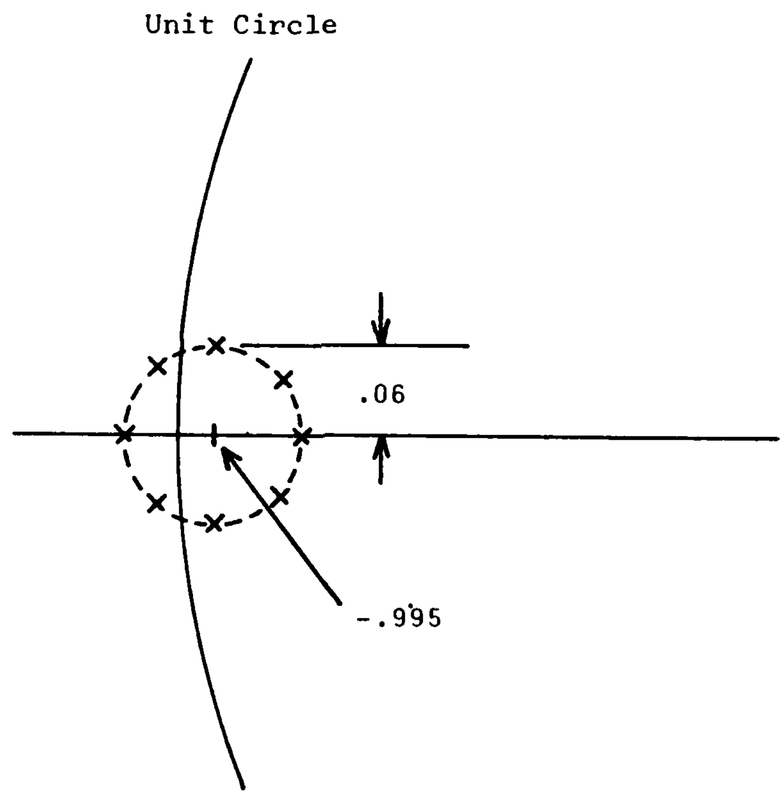


Figure 8-2 Three Of These Complex Poles Are Outside The Unit Circle

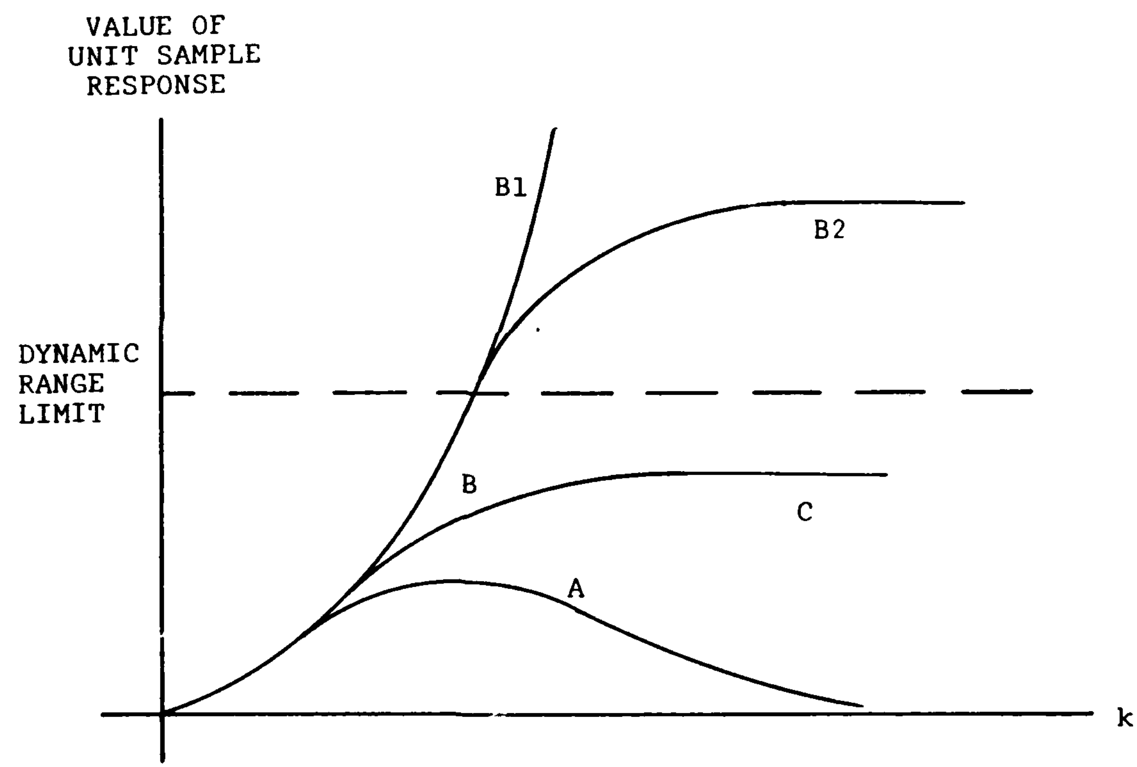


Figure 8-3 Summary Of Possible Observed Unit Sample Responses

IX. SUMMARY AND CONCLUSIONS

The Advanced Simulation of Digital Filters has been implemented on the IBM 360, and is intended for the use by persons beginning their study of digital signal processing. The programs included within the package have been designed to make the user's interaction with the z-plane as simple as possible. While the programs themselves are available for study, all of the computations are invisible to the user, in an attempt to develop the user's intuition of the discrete domain without overwhelming him with arithmetic computations.

The ASDF package provides the user with an interactive method by which the filter pole and zero locations can be manipulated and the effects observed of such movement. ASDF can simulate filters of up to tenth order, and is punctuated with numerous comments and instructions which are to assist the user. The interface with the z-plane is available in a rapid interactive graphics mode, as well as in the slower polar and rectangular modes, each providing greater position location accuracy. The filter responses generated by ASDF are presented as a series of displays directly on the Tektronix 4012 terminal. Versatec plot capability has been integrated into the system to provide high quality copies of the system responses.

A. CONCLUSIONS

There are several points which are worth considering for persons intending to pursue this or other areas of interactive systems development with the graphics terminals. While comments may appear obvious they are stated to minimize the difficulties which can be encountered. First, a serious attempt should be made to define the entire data structure before beginning the programming stage of the work. All phases of the effort must be planned before commencing with the actual programming to insure that the data structure selected is sufficiently general, and sufficiently flexible to endure the additions and changes which inevitably occur when initial objectives are overlooked or changed. Second, it is imperative to master all of the software tools available and applicable to the project. There are features of the PLOT 10 software which exist and were not used in the ASDF program. This shortcoming is due to insufficient a priori knowledge. For example, the writing of text portions of the screen presentations should have been much simpler. Finally, it is important to research all of the attributes of the devices used for the final problem solution, and direct the solution to take advantage of all the system capabilities.

B. AREAS FOR FUTURE WORK

There are several areas in which this program could be improved or extended. The possibility exists for transferring the ASDF programs to the Tektronix 4082 computing system. This move would avoid many of the shortcommings associated with the current IBM 360 system. The user would be relatively immune to system crashes, and the constraints placed upon him by concerning the IBM system's being operational and/or available. There are several disadvantages associated with this move. With the advent of the new computer system the majority of the current problems concerning system reliability, and constraints on available access hours should be eliminated. Furthermore, transferring these programs to the 4082 system will require significant rewriting to make execution possible on a system which has a much smaller virtual machine. Furthermore to fully benefit from the 4082 system, the interfacing routines should be rewritten to take advantage of the 4082 refresh capability. It seems that all of the effort involved for this change is not worth the expected improvements.

The ASDF program is now useful for analyzing digital filters. It seems likely that this capability could be expanded into a method for computer-aided design of digital filters. There are standard transformations which allow the filter designer to map analog filter requirements into the

appropriate digital requirements, and as well there are transformations which make possible the computation of digital filter transfer functions from known analog transfer functions. There are also a variety of optimization techniques referenced in the literature which could be implemented.

The other area of obvious expansion for the ASDF program is to include the capability for performing spectral analysis. It would be worthwhile (though perhaps not particularly involved) to include in the system the ability to calculate Discrete Fourier Transforms, and graphically display the results. Additionally, it would be possible to demonstrate the correspondence between performing convolution in the time domain directly and performing the convolution in the frequency domain by multiplying the DFT's. The relation between the circular and linear convolution could also be graphically portrayed.

As a final precautionary note, it is not possible to overestimate the length of time required to generate well documented, correct, useful programs on an interactive basis. Changes in system parameters, and the limitations in system software can make the process of simplifying the user's interactive responsibilities a very difficult process for anyone not thoroughly acquainted with all available resources. The myriad of possible incorrect user responses which must be anticipated make the problem extensive.

APPENDIX A: PROBABLE SOLUTIONS FOR DIFFICULTIES

SYMPTOM: RESPONSE command returns an error message upon execution.

SOLUTION: Check the virtual machine size with the command "CP Q F" to insure a virtual machine size of 400K. If the user has only been allocated 256K, logoff and log back on as directed in chapter five. Load ASDF as described in chapter six.

SYMPTOM: User has inadvertently entered the control program, ie., CP.

SOLUTION: Type in "b" and "return." Normal program execution should resume.

SYMPTOM: The carat appears with the blinking cursor, but it is not located in the command array.

SOLUTION: Move the trace cursor into the command array, and type a "space." Normal operation should resume.

SYMPTOM: No terminal activity.

SOLUTION: First, check to see whether or not the system has crashed. If the system has crashed, logon as in chapter five, and load ASDF as in chapter six. If the system has not crashed, move the cursor to the center of the screen and

type "space" and "return." If normal operation has not resumed hit the "break" key. When the CP prompt appears, type "b." Normal operation should resume.

SYMPTOM: An error is detected in the execution of ASDF191 EXEC indicating that no temporary disk space is available.
SOLUTION: Check with the consultant or a systems programmer.

SYMPTOM: The user can not escape from a delete pole or delete zero command in the interactive graphics mode (IAG).
SOLUTION: Implicit in each request to delete from the z-plane is a request, validating the change to be made. In the delete IAG mode the trace cursor returns after illustrating the root to be deleted. Type "y" if the correct root was identified. Type "n" if the wrong root was identified, and attempt the procedure again.

SYMPTOM: The user did not enter the correct amount of data and the program terminated on a 217 error.
SOLUTION: Type "b" and "return." The program should restart with the ASDF signature. At this point reenter the command desired. If executing the POLZRO command it is advisable to execute the list (LST) command to check on the current filter status.

APPENDIX B: IMPORTANT SUBROUTINES AND VARIABLES

The subroutines and variables described in this appendix are applicable to all programs within the ASDF package.

A. SUBROUTINE NAMES

The subroutines within the ASDF package are divided into three major catagories: 1) Plot-10 software, 2) Versatec software, and 3) routines generated explicitely for the ASDF programs.

1. Plot-10 Software Subroutines

The Plot-10 subroutines are all adaquately described in reference 10, and are listed below.

ERASE	MOVABS	ANMODE
RECOVR	HOME	LINEF
NEWLIN	SWINDO	VWINDO
POLTRN	MOVEA	DRAWSA
LINTRN	DRAWR	DRWREL
MOVREL	VCURSR	BELL
POINTA	SCURSR	INIT
TABHOR	TABVER	SEELOC
FIN	FINITT	DRAWA
DRWABS		

2. Versatec Subroutines

The Versatec subroutines are listed below, and are explained in detail in reference 7.

PLOTS	PLOT	WINDOW
SCALE	AXIS	SYMBOL
LINE		

3. ASDF Subroutines

Each of the following subroutines have been generated for the ASDF programs.

CRRMMM - Add and delete poles and zeros from the z-plane. The change "C" can be either "A" for add or "D" for delete. The roots "RR" can be "RZ" for real zero, "CZ" for complex zero, "RP" for real pole, or "CP" for complex pole. Mode "MMM" can be "IAG" for interactive graphics, "POL" for poles, or "RCT" for rectangular.

FLTR2 - Prints a listing of the pole and zero locations of the filter being considered.

ASUBK - Generates the a_k coefficients required to evaluate the transfer function.

BSUBR - Generates the b_r coefficients required to evaluate the transfer function.

ADJUST - Insures that the correct number of zeros located at the origin are included in the coefficients a_k and b_r.

RSPNSE - Computes the time responses of the filter.

FRQNCY - Computes the phase and magnitude of the transfer function for the prescribed filter.

PRPPLT - Scales the ordinate data prior to plotting the responses.

PLTIMP - Plots the unit sample response.

PLTSTP - Plots the unit step response.

PLTTRF - Plots the phase of the transfer function as well as the magnitude in linear and decibel format.

UNTCR - Plots the unit circle with the appropriate poles and zeros.

START - Initializes all arrays and required variables.

FINISH - Checks to insure that the constructed filter is causal, and generates a file to save the pole and zero locations.

ANGLE - Draws the angle sign for polar commands.

ERROR - Controls program flow and error messages for user generated errors.

LST - Prints a listing of the current pole zero location.

PLTTBL - Plots the poles and zeros on the unit circle.

CHCKSZ - Insures that roots are available to be deleted and that the maximum system order will not be exceeded.

UPDATE _ Makes changes in the POLZRO array to correspond to user changes in the z-plane.

LOCATE - Searches the POLZRO table for the root closest to that which the user wishes to delete.

STRRTS - Puts poles and zeros into the POLZRO array.

PLOTRT - Plots a single pole or zero on the unit circle.

B. IMPORTANT VARIABLES

The variables described below pertain to all of the subroutines and programs within the ASDF package.

IROOTS(I) - Keeps track of the number of poles and zeros in the system under consideration. IROOTS(1) is the number of real zeros. IROOTS(2) is the number of complex zero pairs. IROOTS(3) is the number of real poles. IROOTS(4) is the number of complex pole pairs.

POLZRO(I,J) - The locations of the zeros are stored for values of I from one to ten. The pole locations are stored for values of I from 11 through 20. The real part of the rectangular representation of a root location is stored with J equal to one. The imaginary part of the rectangular represention for a root location is stored with J equal to two. The magnitude and angle of the root's polar representation are stored for J values of three and four, respectively. POLZRO(I,5) holds the root type.

IERROR - Keeps track of the generation of an error. Once the error has been identified the value is set to one. When the error has been explained to the user the value is reset to zero.

A(I) and B(I) - Store the values of the a_k and b_r coefficients.

AA(I) and BB(I) - Store the double precision values of the a_k and b_r coefficients.

YI(I) - Stores the values of the unit sample response to be plotted.

YS(I) - Stores the values of the unit step response to be plotted.

YP(I) - Stores the values of the phase of the transfer function to be plotted.

YM(I) - Stores the magnitudes of the transfer function to be plotted.

YMDB(I) - Stores the transfer function magnitudes in decibels to be plotted.

EN(I) - Stores the time values against which the dependent variables are plotted.

IYIPTS - Holds the number of unit sample points to be plotted.

IYSPTS - Holds the number of unit step response points to be plotted.

NDEN - Is the order of the denominator plus one.

NNUM - Is the order of the numerator plus one.

NORDER - Is the order of the system under consideration. This can be either of the numberator or denominator.

LAST - Holds the number of points to be plotted.

FLTRGN - Is the value that the user assigns to the filter gain.

ZEROS(I) - Holds the complex coefficients which will be made real to generate the b_r coefficients.

POLES(I) - Holds the complex coefficients which will be made real to generate the a_k coefficients.

The remainder of the variables are either obvious, or used merely for control of the program flow, ie. do loop indicies, etc.

APPENDIX C. STRUCTURE FOR ASDF TEXT FILE

The file ASDF TEXT is a composite file made up of all of the required programs and EXEC files for running the ASDF program package. These files have been consolidated to minimize the time required for transfer from DISK02 to OSX001. The structure of the composite file is listed below:

BREAK001

EXEC file ASDF192

BREAK002

TEXT file for POLZRO command

BREAK003

TEXT file for RESPONSE command

BREAK004

Information printed out by the INTRO command

BREAK005

VTEC1 JCL - used for EXEC HRD$CPY

BREAK006

VTEC2 JCL - used for EXEC HRD$CPY

BREAK007

Lables for RESPONSE plots

BREAK008

EXEC file to execute the EXEC PUNCH command

BREAK009

Initial file FT01F001 with no poles or zeros

BREAK010

TEXT file for generating JOB cards

BREAK011

TEXT file for the INTRO command

BREAK012

Information printed by OPTIONS command

BREAK013

Information printed on the thtle page

BREAK014

EXEC file to execute HRD$CPY command

BREAK015

TEXT file for the OPTIONS command

BREAK016

TEXT file for the software page clear

BREAK017

APPENDIX D. JCL FOR EXEC HRD$CPY COMMAND

The Job Control Language required for the execution of the EXEC HRD$CPY command is provided to give insight into the file manipulation utilized by the ASDF package. Should the permanent locations of any of the portions of the ASDF package change this section of JCL must be changed also.

JCL for VTEC1

```
//GO EXEC PGM=THEPLT,REGION=180K
//STEPLIB DD UNIT=3330,VOL=SER=DISK03,DISP=SHR,DSN=S1502.ASDF
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT15F001 DD DDNAME=PLOTPARM
//PLOTLOG DD SYSOUT=A
//VECTR2 DD DSN=&&VECTR2,DISP=(,PASS),SPACE=(CYL,(5,5)),UNIT=SYSDA
//VECTR1 DD DSN=&&VECTR1,DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSDA
//GO.SYSIN DD *
```

JCL for VTEC2

```
//PLOT   EXEC   PGM=IEVMAPP,COND=(4,LT)
//FT15F001 DD DDNAME=PLOTPARM
//STEPLIB  DD DSN=SYS1.VTECPLOT,DISP=SHR
//PLOTLOG  DD SYSOUT=A
//VECTR1  DD  DISP=(OLD,DELETE),DSN=&&VECTR1
//VECTR2  DD  DISP=(OLD,DELETE),DSN=&&VECTR2
//SYSVECTR  DD  SYSOUT=(A,,5555)
//VECTTAPE  DD  DUMMY
//
```

APPENDIX E: SOURCE DECK FOR ASDF191 EXEC

```
&COMMENT******************************************************
&COMMENT*                                                    *
&COMMENT*     SOURCE DECK FOR ASDF COMMAND:  EXEC ASDF191    *
&COMMENT*                                                    *
&COMMENT******************************************************
&COMMENT
&COMMENT  GENERATE A STACK TO EDIT THE ASDF192 FILE
&COMMENT  OFF OF THE BEGINNING OF THE ASDF TEXT FILE
&COMMENT  WHICH IS STORED ON DISK02
&COMMENT
&BEGSTACK
F0718.ASDF#
N
D
F BREAK002
D 50000
#   FILE ASDF192
&ENDSTACK
&COMMENT
&COMMENT  RESERVE AND FORMAT THE TEMPORARY DISK SPACE
&COMMENT
CP DEFINE T2314 192 10
FORMAT T ALL (NOTYPE)
&COMMENT
&COMMENT  GIVE THE PROGRAMS ACCESS TO ALL REQUIRED LIBRARIES
&COMMENT
GLOBAL T TEKLIB SYSLIB
CP SET LINELN 72
&COMMENT
&COMMENT  MODIFY THE TOCP EXEC ON THE SYSTEM DISK
&COMMENT  SO THAT IT WILL ACCEPT THE STACK OF
&COMMENT  EDIT COMMANDS AND GET ASDF TEXT FROM DISK
&COMMENT
LOADMOD OSDISK (CLEAR
CP ST 138E4 18111811
CP ST 1392C 4780CF34
START * ASDF TEXT T1 237
&COMMENT
&COMMENT  GENERATE THE CORRECT FILETYPE FOR ASDF
&COMMENT
ALTER ASDF TEXT T1 ASDF EXEC T1
&COMMENT
&COMMENT  PERFORM THE EDITING OF THE ASDF TEXT FILE
&COMMENT
CEDIT ASDF EXEC
ALTER ASDF EXEC T1 ASDF TEXT T1
&COMMENT
&COMMENT  PREPARE TO MOVE THE USER TO THE
```

```
&COMMENT  TEMPORARY DISK - INFORM THE USER OF                                ASDF1049
&COMMENT  THE APPROPRIATE COMMANDS TO ISSUE                                  ASDF1050
&COMMENT                                                                     ASDF1051
RELEASE 192 T                                                                ASDF1052
&PRINT                                                                       ASDF1053
&PRINT                                                                       ASDF1054
&PRINT                                                                       ASDF1055
&PRINT ISSUE THE COMMAND:  LOGIN 192 P                                       ASDF1056
&PRINT                                                                       ASDF1057
&PRINT                                                                       ASDF1058
&PRINT                                                                       ASDF1059
&PRINT ISSUE THE COMMAND: EXEC ASDF192                                       ASDF1060
&PRINT                                                                       ASDF1061
&PRINT                                                                       ASDF1062
&PRINT                                                                       ASDF1063
```

APPENDIX F: SOURCE DECK FOR ASDF192 EXEC

```
C*********************************************             ASDF2001
C*                                            *             ASDF2002
C*           SOURCE DECK FOR ASDF192          *             ASDF2003
C*                                            *             ASDF2004
C*********************************************             ASDF2005
&COMMENT                                                    ASDF2006
&COMMENT GENERATE A STACK OF EDIT COMMANDS TO               ASDF2007
&COMMENT PROCESS THE COMBINED ASDF TEXT FILE.               ASDF2008
&COMMENT                                                    ASDF2009
&BEGSTACK                                                   ASDF2010
N                                                           ASDF2011
D                                                           ASDF2012
#    FILE                                                   ASDF2013
N                                                           ASDF2014
D                                                           ASDF2015
#    FILE                                                   ASDF2016
N                                                           ASDF2017
D                                                           ASDF2018
#    FILE                                                   ASDF2019
N                                                           ASDF2020
D                                                           ASDF2021
#    FILE                                                   ASDF2022
N                                                           ASDF2023
D                                                           ASDF2024
#    FILE                                                   ASDF2025
N                                                           ASDF2026
D                                                           ASDF2027
#    FILE                                                   ASDF2028
N                                                           ASDF2029
D                                                           ASDF2030
#    FILE                                                   ASDF2031
N                                                           ASDF2032
D                                                           ASDF2033
#    FILE                                                   ASDF2034
N                                                           ASDF2035
D                                                           ASDF2036
#    FILE                                                   ASDF2037
N                                                           ASDF2038
D                                                           ASDF2039
#    FILE                                                   ASDF2040
N                                                           ASDF2041
D                                                           ASDF2042
#    FILE                                                   ASDF2043
N                                                           ASDF2044
D                                                           ASDF2045
#    FILE                                                   ASDF2046
N                                                           ASDF2047
D                                                           ASDF2048
```

```
#    FILE                                                          ASDF2049
N                                                                  ASDF2050
D                                                                  ASDF2051
#    FILE                                                          ASDF2052
N                                                                  ASDF2053
D                                                                  ASDF2054
#    FILE                                                          ASDF2055
&ENDSTACK                                                          ASDF2056
&COMMENT                                                           ASDF2057
&COMMENT SPLIT EACH SUCCESSIVE FILE OFF OF THE                     ASDF2058
&COMMENT COMPOSITE ASDF TEXT FILE; THEN CHANGE                     ASDF2059
&COMMENT THE FILE TYPE TO P1.                                      ASDF2060
&COMMENT                                                           ASDF2061
SPLIT ASDF TEXT POLZRO TEXT BREAK002 BREAK003                      ASDF2062
ALTER POLZRO TEXT P5 POLZRO TEXT P1                                ASDF2063
SPLIT ASDF TEXT RESPONSE TEXT BREAK003 BREAK004                    ASDF2064
ALTER RESPONSE TEXT P5 RESPONSE TEXT P1                            ASDF2065
SPLIT ASDF TEXT INTRO FT04F001 BREAK004 BREAK005                   ASDF2066
ALTER INTRO FT04F001 P5 INTRO FT04F001 P1                          ASDF2067
SPLIT ASDF TEXT VTEC1 JCL BREAK005 BREAK006                        ASDF2068
ALTER VTEC1 JCL P5 VTEC1 JCL P1                                    ASDF2069
SPLIT ASDF TEXT VTEC2 JCL BREAK006 BREAK007                        ASDF2070
ALTER VTEC2 JCL P5 VTEC2 JCL P1                                    ASDF2071
SPLIT ASDF TEXT FILE FT02F001 BREAK007 BREAK008                    ASDF2072
ALTER FILE FT02F001 P5 FILE FT02F001 P1                            ASDF2073
SPLIT ASDF TEXT  PUNCH EXEC BREAK008 BREAK009                      ASDF2074
ALTER PUNCH EXEC P5 PUNCH EXEC P1                                  ASDF2075
SPLIT ASDF TEXT FILE FT01F001 BREAK009 BREAK010                    ASDF2076
ALTER FILE FT01F001 P5 FILE FT01F001 P1                            ASDF2077
SPLIT ASDF TEXT JOB$CRD TEXT BREAK010 BREAK011                     ASDF2078
ALTER JOB$CRD TEXT P5 JOB$CRD TEXT P1                              ASDF2079
SPLIT ASDF TEXT INTRO TEXT BREAK011 BREAK012                       ASDF2080
ALTER INTRO TEXT P5 INTRO TEXT P1                                  ASDF2081
SPLIT ASDF TEXT OPTIONS FT03F001 BREAK012 BREAK013                 ASDF2082
ALTER OPTIONS FT03F001 P5 FILE FT03F001 P1                         ASDF2083
SPLIT ASDF TEXT COVER FT04F001 BREAK013 BREAK014                   ASDF2084
ALTER COVER FT04F001 P5 COVER FT04F001 P1                          ASDF2085
SPLIT ASDF TEXT HRD$CPY EXEC BREAK014 BREAK015                     ASDF2086
ALTER HRD$CPY EXEC P5 HRD$CPY EXEC P1                              ASDF2087
SPLIT ASDF TEXT OPTIONS TEXT BREAK015 BREAK016                     ASDF2088
ALTER OPTIONS TEXT P5 OPTIONS TEXT P1                              ASDF2089
SPLIT ASDF TEXT CLEAR TEXT BREAK016 BREAK017                       ASDF2090
ALTER CLEAR TEXT P5 CLEAR TEXT P1                                  ASDF2091
&COMMENT                                                           ASDF2092
&COMMENT EDIT EACH OF THE FILES JUST CONSTRUCTED                   ASDF2093
&COMMENT TO ELIMINATE THE LOGICAL SEPARATORS.                      ASDF2094
&COMMENT THESE SEPARATORS ARE                                      ASDF2095
&COMMENT ALL OF THE FORM BREAKXXX.                                 ASDF2096
```

```
&COMMENT                                         ASDF2097
EDIT POLZRO TEXT P1                              ASDF2098
EDIT RESPONSE TEXT P1                            ASDF2099
EDIT INTRO FT04F001 P1                           ASDF2100
EDIT VTEC1 JCL P1                                ASDF2101
EDIT VTEC2 JCL P1                                ASDF2102
EDIT FILE FT02F001 P1                            ASDF2103
EDIT PUNCH EXEC P1                               ASDF2104
EDIT FILE FT01F001 P1                            ASDF2105
EDIT JOB$CRD TEXT P1                             ASDF2106
EDIT INTRO TEXT P1                               ASDF2107
EDIT FILE FT03F001 P1                            ASDF2108
EDIT COVER FT04F001 P1                           ASDF2109
EDIT HRD$CPY EXEC P1                             ASDF2110
EDIT OPTIONS TEXT P1                             ASDF2111
EDIT CLEAR TEXT P1                               ASDF2112
&COMMENT LOAD EACH OF THE CREATED FILES          ASDF2113
&COMMENT AND GENERATE A CORE IMAGE               ASDF2114
&COMMENT SO THAT SUBSEQUENT LOADING AND          ASDF2115
&COMMENT LINKING IS NOT REQUIRED.                ASDF2116
&COMMENT                                         ASDF2117
LOAD POLZRO                                      ASDF2118
GENMOD POLZRO (NOMAP P2)                         ASDF2119
LOAD RESPONSE                                    ASDF2120
GENMOD RESPONSE (NOMAP P2)                       ASDF2121
LOAD INTRO                                       ASDF2122
GENMOD INTRO                                     ASDF2123
LOAD JOB$CRD                                     ASDF2124
GENMOD JOB$CRD                                   ASDF2125
LOAD OPTIONS                                     ASDF2126
GENMOD OPTIONS (NOMAP P2)                        ASDF2127
LOAD CLEAR                                       ASDF2128
GENMOD CLEAR (NOMAP P2)                          ASDF2129
&TYPEOUT OFF                                     ASDF2130
ALTER COVER FT04F001 P1 FILE FT04F001 P1         ASDF2131
INTRO                                            ASDF2132
ALTER FILE FT04F001 P1 COVER FT04F001 P1         ASDF2133
ALTER INTRO FT04F001 P1 FILE FT04F001 P1         ASDF2134
OPTIONS                                          ASDF2135
CLEAR                                            ASDF2136
&COMMENT                                         ASDF2137
&COMMENT THESE NEXT STATEMENTS FORM AN           ASDF2138
&COMMENT INFINITE LOOP WHICH PROCESSES           ASDF2139
&COMMENT SUCCESSIVE ASDF AND CP/CMS COMMANDS.    ASDF2140
&COMMENT                                         ASDF2141
-LOOP &PRINT *ASDF-READY                         ASDF2142
&READ                                            ASDF2143
&ERROR &GOTO -LOOP2                              ASDF2144
```

```
CLEAR                                                         ASDF2145
&GOTO -LOOP                                                   ASDF2146
-LOOP2 &PRINT INPUT COMMAND ERROR                             ASDF2147
&GOTO -LOOP                                                   ASDF2148
```

```
C*********************************************                           ASDF3001
C*                                           *                           ASDF3002
C*     SOURCE DECK FOR ASDF COMMAND: POLZRO  *                           ASDF3003
C*                                           *                           ASDF3004
C*********************************************                           ASDF3005
C*                                                                       ASDF3006
C* THIS IS THE MAIN PROGRAM FROM WHICH ALL SUBROUTINES                   ASDF3007
C* WILL BE CALLED                                                        ASDF3008
C*                                                                       ASDF3009
      COMMON IROOTS(4),POLZRO(20,5),IERROR,IOLD                          ASDF3010
      CALL START                                                         ASDF3011
 10   CALL UNTCIR                                                        ASDF3012
      IF(IROOTS(1)+2*IROOTS(2).LE.IROOTS(3)+2*IROOTS(4))CALL FINISH      ASDF3013
      IF(IROOTS(1)+2*IROOTS(2).LE.IROOTS(3)+2*IROOTS(4))STOP             ASDF3014
      CALL ERASE                                                         ASDF3015
      CALL MOVABS(126,425)                                               ASDF3016
      CALL ANMODE                                                        ASDF3017
      WRITE(6,100)                                                       ASDF3018
 100  FORMAT(' FILTER IS NOT CAUSAL, TAKE APPROPRIATE CORRECTIVE',       ASDF3019
     X' ACTION.')                                                        ASDF3020
      CALL RECOVR                                                        ASDF3021
      CALL MOVABS(301,381)                                               ASDF3022
      CALL ANMODE                                                        ASDF3023
      WRITE(6,102)                                                       ASDF3024
 102  FORMAT(' HIT SPACE AND RETURN TO CONTINUE')                        ASDF3025
      CALL RECOVR                                                        ASDF3026
      CALL MOVABS(525,337)                                               ASDF3027
      CALL ANMODE                                                        ASDF3028
      READ(5,101) IN                                                     ASDF3029
 101  FORMAT(A1)                                                         ASDF3030
      CALL RECOVR                                                        ASDF3031
      IOLD=1                                                             ASDF3032
      GO TO 10                                                           ASDF3033
      END                                                                ASDF3034
                                                                         ASDF3035
      SUBROUTINE LETTER                                                  ASDF3036
C*                                                                       ASDF3037
C* THIS SUBROUTINE GENERATES THE ALPHANUMERIC INFORMATION FOR            ASDF3038
C* THE UNIT CIRCLE SUBROUTINE  (32)                                      ASDF3039
C*                                                                       ASDF3040
      DIMENSION ICMD(13)                                                 ASDF3041
      DATA ICMD/3HACZ,3HACP,3HARZ,3HARP,                                 ASDF3042
     1    3HDCZ,3HDCP,3HDRZ,3HDRP,3HNUC,                                 ASDF3043
     2    3HLST,3HEEM,3HRDY,3HEXC/                                       ASDF3044
C*                                                                       ASDF3045
C* MOVE THE CURSOR TO THE SCREEN TOP LEFT CORNER  (1)                    ASDF3046
C*                                                                       ASDF3047
      CALL HOME                                                          ASDF3048
```

```
C*
C* GENERATE A BLANK LINE  (1)
C*
      CALL LINEF
C*
C* ANMODE IS REQUIRED FOR FORTRAN I/O IN GRAPHIC MODE  (1)
C*
      CALL ANMODE
C*
C* PRINT THE TITLES  (2)
C*
      WRITE (6,100)
 100  FORMAT (6X,20HUNIT CIRCLE COMMANDS)
C*
C* GET THE TERMINAL BACK TO THE GRAPHIC MODE  (1)
C*
      CALL RECOVR
      CALL NEWLIN
      CALL LINEF
      CALL ANMODE
      WRITE (6,101)
 101  FORMAT (9X,18HINTER-    KEYBOARD)
      CALL RECOVR
C*
C* GENERATE A CARRIAGE RETURN AND A LINE FEED  (1)
C*
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,102)
 102  FORMAT (3X,25HCMDS  ACTIVE  RECT  POLAR)
      CALL RECOVR
      CALL NEWLIN
      CALL LINEF
C*
C* PRINT THE COMMANDS (6)
C*
      DO 40 M=1,10
      CALL ANMODE
      WRITE (6,103) ICMD(M)
      CALL RECOVR
 40   CALL LINEF
 103  FORMAT (4X,A3)
      RETURN
      END

      SUBROUTINE PLOTUC
C*
C* THIS SUBROUTINE DRAWS THE UNIT CIRCLE WITH THE AXES
```

```
C*                                                                      ASDF3097
C* CONFINE THE UNIT CIRCLE TO A 600 X 600 RASTER UNIT SQUARE           ASDF3098
C* POSITION THE SCREEN WINDOW                                           ASDF3099
C* XMIN = (1023-600) = 423                                              ASDF3100
C* YMIN = 0                                                             ASDF3101
C*                                                                      ASDF3102
      CALL SWINDO (393,630,0,630)                                       ASDF3103
C*                                                                      ASDF3104
C* GENERATE THE UNIT CIRCLE  (4)                                        ASDF3105
C*                                                                      ASDF3106
      CALL VWINDO (0.,1.1,0.,360.)                                      ASDF3107
      CALL POLTRN (0.,360.,0.)                                          ASDF3108
      CALL MOVEA (1.,0.)                                                ASDF3109
      CALL DRAWSA (1.,360.)                                             ASDF3110
C*                                                                      ASDF3111
C* RETURN TO RECTANGULAR COORDINATES                                    ASDF3112
C*  VIRTUAL WINDOW: -1.1 LE X,Y LE +1.1 (2)                             ASDF3113
C*                                                                      ASDF3114
      CALL LINTRN                                                       ASDF3115
      CALL VWINDO (-1.1,2.2,-1.1,2.2)                                   ASDF3116
C*                                                                      ASDF3117
C* GENERATE THE X-AXIS  (2)                                             ASDF3118
C*                                                                      ASDF3119
      CALL MOVEA (1.,0.)                                                ASDF3120
      CALL DRAWR (-2.,0.)                                               ASDF3121
C*                                                                      ASDF3122
C* GENERATE THE Y-AXIS  (2)                                             ASDF3123
C*                                                                      ASDF3124
      CALL MOVEA (0.,1.)                                                ASDF3125
      CALL DRAWR (0.,-2.)                                               ASDF3126
C*                                                                      ASDF3127
C* GENERATE THE X-AXIS TICK MARKS  (5)                                  ASDF3128
C* B IS THE TICK MARK LENGTH IN USER UNITS                              ASDF3129
C*                                                                      ASDF3130
      DO 10 I=1,19                                                      ASDF3131
      B=.05                                                             ASDF3132
      IF(I.EQ.5.OR.I.EQ.15)B=.1                                         ASDF3133
      CALL MOVEA (-1.+FLOAT(I)*.1,-B/2.)                                ASDF3134
 10   CALL DRAWR (0.,B)                                                 ASDF3135
C*                                                                      ASDF3136
C* GENERATE THE Y-AXIS TICK MARKS  (5)                                  ASDF3137
C* B IS THE TICK MARK LENGTH IN USER UNITS                              ASDF3138
C*                                                                      ASDF3139
      DO 20 I=1,19                                                      ASDF3140
      B=.05                                                             ASDF3141
      IF(I.EQ.5.OR.I.EQ.15)B=.1                                         ASDF3142
      CALL MOVEA (-B/2.,1.-FLOAT(I)*.1)                                 ASDF3143
 20   CALL DRAWR (B,0.)                                                 ASDF3144
```

```
      RETURN                                                            ASDF3145
      END                                                               ASDF3146
                                                                        ASDF3147
      SUBROUTINE BOXUC                                                  ASDF3148
C*                                                                      ASDF3149
C* THE PURPOSE OF THIS SUBROUTINE IS TO GENERATE THE "CHOICE"           ASDF3150
C* BOXES FOR THE UNIT CIRCLE SUBROUTINE                                 ASDF3151
C*                                                                      ASDF3152
      DIMENSION IBCORD (3,3)                                            ASDF3153
C*                                                                      ASDF3154
C* DATA DESCRIBING THE LOWER LEFT CORNER (LLC) OF                       ASDF3155
C* EACH BOX AND THE NUMBER OF BOXES IN EACH COLUMN                      ASDF3156
C*       IBCORD (J,1)  IBCORD (J,2)  IBCORD (J,3)                       ASDF3157
C*           LLCX           LLCY         NO. BOXES                      ASDF3158
C*                                                                      ASDF3159
      DATA IBCORD/140,238,336,439,483,483,10,8,8/                       ASDF3160
C*                                                                      ASDF3161
C* GENERATE THE "CHOICE" BOXES  (10)                                    ASDF3162
C*                                                                      ASDF3163
      DO 10 J=1,3                                                       ASDF3164
C*                                                                      ASDF3165
C* MOVE TO THE LOWER LEFT CORNER OF THE BOX  (1)                        ASDF3166
C*                                                                      ASDF3167
      CALL MOVABS (IBCORD(J,1),IBCORD(J,2))                             ASDF3168
C*                                                                      ASDF3169
C*  "IBCORD(J,3)" CONTROLS THE NUMBER OF BLOCKS TO BE DRAWN (10)        ASDF3170
C*                                                                      ASDF3171
      K=IBCORD(J,3)                                                     ASDF3172
      DO 20 I=1,K                                                       ASDF3173
C*                                                                      ASDF3174
C* DRAW THE ACTUAL BOX  (4)                                             ASDF3175
C*                                                                      ASDF3176
      CALL DRWREL (0,16)                                                ASDF3177
      CALL DRWREL (14,0)                                                ASDF3178
      CALL DRWREL (0,-16)                                               ASDF3179
      CALL DRWREL (-14,0)                                               ASDF3180
C*                                                                      ASDF3181
C* PROCEED TO THE NEXT HIGHER BOX IN THE COLUMN  (1)                    ASDF3182
C*                                                                      ASDF3183
 20   CALL MOVREL (0,22)                                                ASDF3184
 10   CONTINUE                                                          ASDF3185
      RETURN                                                            ASDF3186
      END                                                               ASDF3187
                                                                        ASDF3188
      SUBROUTINE ACZIAG                                                 ASDF3189
      COMMON  IROOTS(4),POLZRO(20,5), IERROR                            ASDF3190
      CALL CHCKSZ (2,2)                                                 ASDF3191
      IF(IERROR.NE.0)RETURN                                             ASDF3192
```

```
      CALL VCURSR (ICHAR,X,Y)
      CALL STRRTS (2,1,X,Y)
      RETURN
      END

      SUBROUTINE ACPIAG
      COMMON IXXXXX(4),PXXXXX(20,5), IERROR
      CALL CHCKSZ (2,4)
      IF(IERROR.NE.0)RETURN
      CALL VCURSR (ICHAR, X,Y)
C*
C* INSURE THAT THE POLES ARE WITHIN THE UNIT CIRCLE
C*
      IF(SQRT(X**2+Y**2).GT.1)IERROR=9
      IF(IERROR.NE.0)RETURN
      CALL STRRTS (4,1,X,Y)
      RETURN
      END

      SUBROUTINE ARZIAG
      COMMON IXXXXX(4),PXXXXX(20,5),IERROR
      CALL CHCKSZ(1,1)
      IF(IERROR.NE.0)RETURN
      CALL VCURSR(ICHAR,X,Y)
      Y = 0
      CALL STRRTS (1,1,X,Y)
      RETURN
      END

      SUBROUTINE ARPIAG
      COMMON IXXXXX(4), PXXXXX(20,5), IERROR
      CALL CHCKSZ (1,3)
      IF(IERROR.NE.0)RETURN
      CALL VCURSR (ICHAR, X, Y)
C*
C* INSURE THAT THE POLE IS WITHIN THE UNIT CIRCLE
C*
      Y=0
      IF(SQRT(X**2+Y**2).GT.1)IERROR=9
      IF(IERROR.NE.0)RETURN
      CALL STRRTS (3,1,X,Y)
      RETURN
      END

      SUBROUTINE DCZIAG
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      CALL CHCKSZ(-1,2)
      IF(IERROR.NE.0)RETURN
```

```
10    CALL VCURSR(ICHAR,X,Y)                                          ASDF3241
      CALL LOCATE(2,X,Y,LINEPZ)                                       ASDF3242
C*                                                                    ASDF3243
C* INDICATE THE COMPLEX ZERO NEAREST THE USER'S CURSOR                ASDF3244
C* WITH A BRIGHT DOT FIVE TIMES  (4)                                  ASDF3245
C*                                                                    ASDF3246
      XX=POLZRO(LINEPZ,1)                                             ASDF3247
      YY=POLZRO(LINEPZ,2)                                             ASDF3248
      IF(POLZRO(LINEPZ,3).LE.1.0)GO TO 15                             ASDF3249
      ANGLE=ATAN2(YY,XX)                                              ASDF3250
      XX=1.05*COS(ANGLE)                                              ASDF3251
      YY=1.05*SIN(ANGLE)                                              ASDF3252
15    DO 20 NREPS=1,5                                                 ASDF3253
      CALL BELL                                                       ASDF3254
      IF(Y.LE.0.)CALL POINTA(XX,-YY)                                  ASDF3255
20    IF(Y.GT.0.)CALL POINTA(XX,YY)                                   ASDF3256
      CALL SCURSR(ICHAR,IX,IY)                                        ASDF3257
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                        ASDF3258
      CALL UPDATE(2,LINEPZ)                                           ASDF3259
      CALL NUC                                                        ASDF3260
      RETURN                                                          ASDF3261
      END                                                             ASDF3262
                                                                      ASDF3263
      SUBROUTINE DCPIAG                                               ASDF3264
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                            ASDF3265
      CALL CHCKSZ(-1,4)                                               ASDF3266
      IF(IERROR.NE.0)RETURN                                           ASDF3267
10    CALL VCURSR (ICHAR,X,Y)                                         ASDF3268
      CALL LOCATE(4,X,Y,LINEPZ)                                       ASDF3269
C*                                                                    ASDF3270
C* INDICATE THE COMPLEX POLE NEAREST THE USER'S CURSOR                ASDF3271
C* FIVE TIMES WITH A BRIGHT DOT  (4)                                  ASDF3272
C*                                                                    ASDF3273
      DO 20 NREPS=1,5                                                 ASDF3274
      CALL BELL                                                       ASDF3275
      IF(Y.LE.0.)CALL POINTA(POLZRO(LINEPZ,1),-POLZRO(LINEPZ,2))      ASDF3276
20    IF(Y.GT.0.)CALL POINTA(POLZRO(LINEPZ,1),POLZRO(LINEPZ,2))       ASDF3277
      CALL SCURSR (ICHAR,IX,IY)                                       ASDF3278
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                        ASDF3279
      CALL UPDATE (4,LINEPZ)                                          ASDF3280
      CALL NUC                                                        ASDF3281
      RETURN                                                          ASDF3282
      END                                                             ASDF3283
                                                                      ASDF3284
      SUBROUTINE DRZIAG                                               ASDF3285
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                            ASDF3286
      CALL CHCKSZ (-1,1)                                              ASDF3287
      IF(IERROR.NE.0)RETURN                                           ASDF3288
```

```
10    CALL VCURSR(ICHAR,X,Y)                                    ASDF3289
      CALL LOCATE(1,X,0.,LINEPZ)                                ASDF3290
      DO 20 NREPS=1,5                                           ASDF3291
      CALL BELL                                                 ASDF3292
20    CALL POINTA(POLZRO(LINEPZ,1),0.)                          ASDF3293
      CALL SCURSR (ICHAR,IX,IY)                                 ASDF3294
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                  ASDF3295
      CALL UPDATE(1,LINEPZ)                                     ASDF3296
      CALL NUC                                                  ASDF3297
      RETURN                                                    ASDF3298
      END                                                       ASDF3299
                                                                ASDF3300
      SUBROUTINE DRPIAG                                         ASDF3301
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                      ASDF3302
      CALL CHCKSZ(-1,3)                                         ASDF3303
      IF(IERROR.NE.0)RETURN                                     ASDF3304
10    CALL VCURSR(ICHAR,X,Y)                                    ASDF3305
      CALL LOCATE(3,X,0.,LINEPZ)                                ASDF3306
      DO 20 NREPS=1,5                                           ASDF3307
      CALL BELL                                                 ASDF3308
20    CALL POINTA(POLZRO(LINEPZ,1),0.)                          ASDF3309
      CALL SCURSR(ICHAR,IX,IY)                                  ASDF3310
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                  ASDF3311
      CALL UPDATE(3,LINEPZ)                                     ASDF3312
      CALL NUC                                                  ASDF3313
      RETURN                                                    ASDF3314
      END                                                       ASDF3315
                                                                ASDF3316
      SUBROUTINE UNTCIR                                         ASDF3317
C*                                                              ASDF3318
C* THIS SUBROUTINE CONTROLS THE IMPLEMENTATION OF               ASDF3319
C* COMMANDS ASSOCIATED WITH THE UNIT CIRCLE                     ASDF3320
C*                                                              ASDF3321
      DATA ISE/2H E/                                            ASDF3322
      DATA IX/2H X/                                             ASDF3323
      DIMENSION IBCORD(4,3),IHTBL(10),IVTBL(10)                 ASDF3324
      COMMON IXXXXX(4),PXXXXX(20,5),IERROR,IOLD                 ASDF3325
C*                                                              ASDF3326
C* SET THE TAB TABLE SIZE TO TEN                                ASDF3327
C*                                                              ASDF3328
      CALL TTBLSZ(10)                                           ASDF3329
C*                                                              ASDF3330
C* LOCATE THE HORIZONTAL AND VERTICAL TABS  (2)                 ASDF3331
C*                                                              ASDF3332
      DATA IHTBL/128,226,324,0,0,0,0,0,0,0/                     ASDF3333
      DATA IVTBL/439,461,483,505,527,549,571,593,615,637/       ASDF3334
C*                                                              ASDF3335
C* INITIATE THE GRAPHICS ROUTINES  (1)                          ASDF3336
```

```
C*                                                                      ASDF3337
      IF(IOLD.EQ.0)CALL INIT                                            ASDF3338
C*                                                                      ASDF3339
C* IF WORKING ON AN OLD FILTER - CONTINUE WITH IT                        ASDF3340
C*                                                                      ASDF3341
      IF(IOLD.EQ.1)CALL NUC                                             ASDF3342
      IF(IOLD.EQ.1)GO TO 30                                             ASDF3343
      IF(IOLD.EQ.0)CALL OLFLTR                                          ASDF3344
      IF(IOLD.EQ.0)CALL ERASE                                           ASDF3345
      IF(IOLD.EQ.1)CALL NUC                                             ASDF3346
      IF(IOLD.EQ.1)GO TO 30                                             ASDF3347
C*                                                                      ASDF3348
C* GENERATE THE ALPHANUMERIC I/O FOR THE COMMAND TABLE                   ASDF3349
C* OF THE UNIT CIRCLE SUBROUTINE  (1)                                    ASDF3350
C*                                                                      ASDF3351
      CALL LETTER                                                       ASDF3352
C*                                                                      ASDF3353
C* GENERATE THE "CHOICE" BOXES  (1)                                      ASDF3354
C*                                                                      ASDF3355
      CALL BOXUC                                                        ASDF3356
C*                                                                      ASDF3357
C* GENERATE THE UNIT CIRCLE  (1)                                         ASDF3358
C*                                                                      ASDF3359
      CALL PLOTUC                                                       ASDF3360
C*                                                                      ASDF3361
C* ALLOW THE USER TO SELECT THE DESIRED COMMAND  (1)                     ASDF3362
C*                                                                      ASDF3363
 30   CALL BELL                                                         ASDF3364
      CALL SCURSR (IIS,IIX,IIY)                                         ASDF3365
C*                                                                      ASDF3366
C* ADJUST CURSOR FOR INACCURACY ON USER'S PART  (1)                      ASDF3367
C*                                                                      ASDF3368
      CALL MOVABS (IIX-63,IIY+3)                                        ASDF3369
C*                                                                      ASDF3370
C* PUT THE ALPHANUMERIC CURSOR IN THE NEAREST BOX  (2)                   ASDF3371
C*                                                                      ASDF3372
      CALL TABHOR(IHTBL)                                                ASDF3373
      CALL TABVER(IVTBL)                                                ASDF3374
C*                                                                      ASDF3375
C* RECORD THE LOCATION OF THE CURSOR AFTER                               ASDF3376
C* THE TAB COMMAND HAS BEEN EXECUTED  (1)                                ASDF3377
C*                                                                      ASDF3378
      CALL SEELOC (IA,IB)                                               ASDF3379
C*                                                                      ASDF3380
C* HIGHLIGHT THE CHOSEN BLOCK / COMMAND WITH THE FLASHING CURSOR  (4)    ASDF3381
C*                                                                      ASDF3382
      CALL ANMODE                                                       ASDF3383
      READ (5,104) IZ                                                   ASDF3384
```

```
 104  FORMAT(A2)                                                              ASDF3385
      CALL RECOVR                                                             ASDF3386
C*                                                                            ASDF3387
C* IF HE HAS RESPONDED WITH "SPACE" AND "X"                                   ASDF3388
C* TERMINATE PROGRAM EXECTUION  (2)                                           ASDF3389
C*                                                                            ASDF3390
      IF(IZ.EQ.IX)RETURN                                                      ASDF3391
C*                                                                            ASDF3392
C* IF THE USER HAS NOT RESPONDED WITH A "SPACE"                               ASDF3393
C* AND AN "E" THEN DO NOT EXECUTE THIS COMMAND                                ASDF3394
C* RETURN HIM TO THE CURSOR  (1)                                              ASDF3395
C*                                                                            ASDF3396
      IF(IZ.NE.ISE)GO TO 30                                                   ASDF3397
C*                                                                            ASDF3398
C* ROUTE THE PROGRAM CONTROL TO THE APPROPRIATE  (34)                         ASDF3399
C* SUBROUTINE BASED ON THE TABBED CURSOR LOCATION                             ASDF3400
C*                                                                            ASDF3401
 40   IF(IA.EQ.128.AND.IB.EQ.637)CALL ACZIAG                                  ASDF3402
      IF(IA.EQ.128.AND.IB.EQ.615)CALL ACPIAG                                  ASDF3403
      IF(IA.EQ.128.AND.IB.EQ.593)CALL ARZIAG                                  ASDF3404
      IF(IA.EQ.128.AND.IB.EQ.571)CALL ARPIAG                                  ASDF3405
      IF(IA.EQ.128.AND.IB.EQ.549)CALL DCZIAG                                  ASDF3406
      IF(IA.EQ.128.AND.IB.EQ.527)CALL DCPIAG                                  ASDF3407
      IF(IA.EQ.128.AND.IB.EQ.505)CALL DRZIAG                                  ASDF3408
      IF(IA.EQ.128.AND.IB.EQ.483)CALL DRPIAG                                  ASDF3409
      IF(IA.EQ.128.AND.IB.EQ.461)CALL NUC                                     ASDF3410
      IF(IA.EQ.128.AND.IB.EQ.439)CALL LST                                     ASDF3411
      IF(IA.EQ.226.AND.IB.EQ.637)CALL ACZRCT                                  ASDF3412
      IF(IA.EQ.226.AND.IB.EQ.615)CALL ACPRCT                                  ASDF3413
      IF(IA.EQ.226.AND.IB.EQ.593)CALL ARZRCT                                  ASDF3414
      IF(IA.EQ.226.AND.IB.EQ.571)CALL ARPRCT                                  ASDF3415
      IF(IA.EQ.226.AND.IB.EQ.549)CALL DCZRCT                                  ASDF3416
      IF(IA.EQ.226.AND.IB.EQ.527)CALL DCPRCT                                  ASDF3417
      IF(IA.EQ.226.AND.IB.EQ.505)CALL DRZRCT                                  ASDF3418
      IF(IA.EQ.226.AND.IB.EQ.483)CALL DRPRCT                                  ASDF3419
      IF(IA.EQ.324.AND.IB.EQ.637)CALL ACZPLR                                  ASDF3420
      IF(IA.EQ.324.AND.IB.EQ.615)CALL ACPPLR                                  ASDF3421
      IF(IA.EQ.324.AND.IB.EQ.593)CALL ARZPLR                                  ASDF3422
      IF(IA.EQ.324.AND.IB.EQ.571)CALL ARPPLR                                  ASDF3423
      IF(IA.EQ.324.AND.IB.EQ.549)CALL DCZPLR                                  ASDF3424
      IF(IA.EQ.324.AND.IB.EQ.527)CALL DCPPLR                                  ASDF3425
      IF(IA.EQ.324.AND.IB.EQ.505)CALL DRZPLR                                  ASDF3426
      IF(IA.EQ.324.AND.IB.EQ.483)CALL DRPPLR                                  ASDF3427
      IF(IERROR.NE.0)CALL ERROR                                               ASDF3428
      GO TO 30                                                                ASDF3429
      END                                                                     ASDF3430
                                                                              ASDF3431
      SUBROUTINE OLFLTR                                                       ASDF3432
```

```
      COMMON IROOTS(4),POLZRO(20,5),IERROR,IOLD,FLTRGN
      CALL MOVABS(126,374)
      CALL ANMODE
      WRITE(6,100)
100   FORMAT(' DO YOU HAVE AN OLD FILTER THAT YOU WISH TO MODIFY?')
      CALL RECOVR
      CALL MOVABS(399,330)
      CALL ANMODE
      WRITE(6,101)
101   FORMAT(' TYPE Y OR N')
      CALL RECOVR
      CALL SCURSR(IN,IX,IY)
      IF(IN.NE.121.AND.IN.NE.89)RETURN
      DO 10 I=1,20
10    READ(1,104) (POLZRO(I,J),J=1,5)
104   FORMAT(1X,5F12.8)
      READ(1,105) (IROOTS(I),I=1,4)
105   FORMAT(1X,4I2)
      READ(1,106) FLTRGN
106   FORMAT(1X,F16.8)
      IOLD=1
      RETURN
      END

      SUBROUTINE FINISH
      COMMON IROOTS(4),POLZRO(20,5),IERROR,IOLD,FLTRGN
      CALL ERASE
      CALL ANMODE
      REWIND 1
      DO 10 I=1,20
      WRITE(1,100) (POLZRO(I,J), J=1,5)
100   FORMAT(1X,5F12.8)
10    CONTINUE
      WRITE(1,101) (IROOTS(I),I=1,4)
101   FORMAT(1X,4I2)
      IF(IOLD.EQ.0)FLTRGN=1.
      CALL RECOVR
      CALL MOVABS(217,528)
      CALL ANMODE
      WRITE(6,102) FLTRGN
102   FORMAT(' THE FILTER GAIN IS CURRENTLY:  ',F12.8)
      CALL RECOVR
      CALL MOVABS(252,484)
      CALL ANMODE
      WRITE(6,103)
103   FORMAT(' DO YOU WISH TO CHANGE THE FILTER GAIN?')
      CALL RECOVR
      CALL MOVABS(427,440)
```

```
      CALL ANMODE
      WRITE(6,104)
104   FORMAT(' (TYPE Y OR N)')
      CALL RECOVR
      CALL SCURSR(NEWGN,IX,IY)
      CALL ANMODE
      IF(NEWGN.NE.121.AND.NEWGN.NE.89)GO TO 30
      CALL RECOVR
      CALL MOVABS(182,396)
      CALL ANMODE
      WRITE(6,106)
106   FORMAT(' TYPE IN THE NEW FILTER GAIN, 12 OR FEWER NUMBERS')
      CALL RECOVR
      CALL MOVABS(357,374)
      CALL ANMODE
      WRITE(6,109)
109   FORMAT(' INCLUDE A DECIMAL POINT')
      CALL RECOVR
      CALL MOVABS(421,330)
      CALL ANMODE
      READ(5,107) FLTRGN
107   FORMAT(F12.8)
      CALL RECOVR
      CALL MOVABS(259,286)
      CALL ANMODE
      WRITE(6,108) FLTRGN
108   FORMAT(' NEW FILTER GAIN IS:  ',F16.8)
110   FORMAT(1X,F16.8)
30    CALL ANMODE
      WRITE(1,110) FLTRGN
      CALL FIN
40    RETURN
      END

      SUBROUTINE NUC
      CALL ERASE
      CALL LETTER
      CALL BOXUC
      CALL PLOTUC
      CALL PLTTBL
      RETURN
      END

      SUBROUTINE ACZRCT
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      DIMENSION LTRACZ(64)
      DATA LTRACZ/4HTO A,4HDD A,4H COM,4HPLEX,4H ZER,4HO TO,4H THE
     X    ,4H    ,4HDISP,4HLAY,,4H ENT,4HER T,4HHE R,4HEAL ,4HAND
```

```
     X     ,4H    ,4HIMAG,4HINAR,4HY PA,4HRTS ,4HWITH,4HIN T,4HHE B       ASDF3529
     X     ,4HOXES,4HPROV,4HIDED,4H. EA,4HCH N,4HUMBE,4HR RE,4HQURI       ASDF3530
     X     ,4HES A,4HDECI,4HMAL.,4H REA,4HL PA,4HRT M,4HINUS,4H SIG       ASDF3531
     X     ,4HNS  ,4HMUST,4H BE ,4HINCL,4HUDED,4H WIT,4HHIN ,4HTHE        ASDF3532
     X     ,4HBOX.,4HNOTE,4H: AL,4HL CO,4HMPLE,4HX ZE,4HROES,4H MUS       ASDF3533
     X     ,4HT BE,4HWITH,4HIN T,4HEN U,4HNITS,4H OF ,4HTHE ,4HORIG/      ASDF3534
      DATA LTRACZ(64)/4HIN. /                                             ASDF3535
      CALL CHCKSZ(2,2)                                                    ASDF3536
      IF(IERROR.NE.0)RETURN                                               ASDF3537
 10   CALL ERASE                                                          ASDF3538
      CALL MOVABS(0,528)                                                  ASDF3539
      DO 30 I=1,57,8                                                      ASDF3540
      CALL ANMODE                                                         ASDF3541
      WRITE(6,100)(LTRACZ(I+J-1),J=1,8)                                   ASDF3542
      CALL RECOVR                                                         ASDF3543
 30   CALL NEWLIN                                                         ASDF3544
 100  FORMAT (21X,8A4)                                                    ASDF3545
      CALL NEWLIN                                                         ASDF3546
      CALL ANMODE                                                         ASDF3547
      WRITE (6,101)                                                       ASDF3548
 101  FORMAT (27X,4HREAL,9X,9HIMAGINARY)                                  ASDF3549
      CALL RECOVR                                                         ASDF3550
      CALL NEWLIN                                                         ASDF3551
      CALL ANMODE                                                         ASDF3552
      WRITE (6,102)                                                       ASDF3553
 102  FORMAT(34X,5H+/-  J)                                                ASDF3554
      CALL RECOVR                                                         ASDF3555
      CALL MOVABS(320,302)                                                ASDF3556
      CALL DRWREL(0,22)                                                   ASDF3557
      CALL DRWREL(138,0)                                                  ASDF3558
      CALL DRWREL(0,-22)                                                  ASDF3559
      CALL DRWREL(-138,0)                                                 ASDF3560
      CALL MOVREL(212,0)                                                  ASDF3561
      CALL DRWREL(0,22)                                                   ASDF3562
      CALL DRWREL(140,0)                                                  ASDF3563
      CALL DRWREL(0,-22)                                                  ASDF3564
      CALL DRWREL(-140,0)                                                 ASDF3565
      CALL MOVREL(0,3)                                                    ASDF3566
      CALL CARTN                                                          ASDF3567
      CALL ANMODE                                                         ASDF3568
      WRITE (6,103)                                                       ASDF3569
 103  FORMAT (22X,'>')                                                    ASDF3570
      READ (5,104) X,Y                                                    ASDF3571
 104  FORMAT(F10.7,5X,F10.7)                                              ASDF3572
      CALL RECOVR                                                         ASDF3573
      IF(SQRT(X**2+Y**2).GE.10.)IERROR=10                                 ASDF3574
      IF(IERROR.NE.0)RETURN                                               ASDF3575
      CALL NEWLIN                                                         ASDF3576
```

```
      CALL NEWLIN                                                       ASDF3577
      CALL ANMODE                                                       ASDF3578
      Z=ABS(Y)                                                          ASDF3579
      WRITE(6,105) X,Z                                                  ASDF3580
105   FORMAT(9X,29HTHE NEW COMPLEX ZERO WILL BE ,F10.7,6H +/- J,F10.7)  ASDF3581
      CALL RECOVR                                                       ASDF3582
      CALL NEWLIN                                                       ASDF3583
      CALL ANMODE                                                       ASDF3584
      WRITE(6,106)                                                      ASDF3585
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF3586
      CALL RECOVR                                                       ASDF3587
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF3588
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF3589
      CALL NUC                                                          ASDF3590
      CALL STRRTS(2,1,X,Y)                                              ASDF3591
      RETURN                                                            ASDF3592
      END                                                               ASDF3593
                                                                        ASDF3594
      SUBROUTINE ACPRCT                                                 ASDF3595
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF3596
      DIMENSION LTRACP(64)                                              ASDF3597
      DATA LTRACP/4HTO A,4HDD A,4H COM,4HPLEX,4H POL,4HE WI,4HTHIN      ASDF3598
     X     ,4H THE,4HUNIT,4H CIR,4HCLE,,4H ENT,4HER T,4HHE R,4HEAL      ASDF3599
     X     ,4HAND ,4HIMAG,4HINAR,4HY PA,4HRTS ,4HWITH,4HIN T,4HHE B     ASDF3600
     X     ,4HOXES,4HPROV,4HIDED,4H. EA,4HCH N,4HUMBE,4HR RE,4HQUIR     ASDF3601
     X     ,4HES A,4HDECI,4HMAL.,4H REA,4HL PA,4HRT M,4HINUS,4H SIG     ASDF3602
     X     ,4HNS  ,4HMUST,4H BE ,4HINCL,4HUDED,4H WIT,4HHIN ,4HTHE      ASDF3603
     X     ,4HBOX.,4HNOTE,4H: AL,4HL CO,4HMPLE,4HX PO,4HLES ,4HMUST     ASDF3604
     X     ,4H BE ,4HLOCA,4HTED ,4HINSI,4HDE T,4HHE U,4HNIT ,4HCIRC/    ASDF3605
      DATA LTRACP(64)/4HLE  /                                           ASDF3606
      CALL CHCKSZ(2,4)                                                  ASDF3607
      IF(IERROR.NE.0)RETURN                                             ASDF3608
10    CALL ERASE                                                        ASDF3609
      CALL MOVABS(0,528)                                                ASDF3610
      DO 30 I=1,57,8                                                    ASDF3611
      CALL ANMODE                                                       ASDF3612
      WRITE(6,100)(LTRACP(I+J-1),J=1,8)                                 ASDF3613
      CALL RECOVR                                                       ASDF3614
30    CALL NEWLIN                                                       ASDF3615
100   FORMAT (21X,8A4)                                                  ASDF3616
      CALL NEWLIN                                                       ASDF3617
      CALL ANMODE                                                       ASDF3618
      WRITE (6,101)                                                     ASDF3619
101   FORMAT (27X,4HREAL,9X,9HIMAGINARY)                                ASDF3620
      CALL RECOVR                                                       ASDF3621
      CALL NEWLIN                                                       ASDF3622
      CALL ANMODE                                                       ASDF3623
      WRITE (6,102)                                                     ASDF3624
```

```
102   FORMAT (34X,5H+/- J)                                              ASDF3625
      CALL RECOVR                                                        ASDF3626
      CALL MOVABS(320,302)                                               ASDF3627
      CALL DRWREL(0,22)                                                  ASDF3628
      CALL DRWREL(138,0)                                                 ASDF3629
      CALL DRWREL(0,-22)                                                 ASDF3630
      CALL DRWREL(-138,0)                                                ASDF3631
      CALL MOVREL(212,0)                                                 ASDF3632
      CALL DRWREL(0,22)                                                  ASDF3633
      CALL DRWREL(140,0)                                                 ASDF3634
      CALL DRWREL(0,-22)                                                 ASDF3635
      CALL DRWREL(-140,0)                                                ASDF3636
      CALL MOVREL(0,3)                                                   ASDF3637
      CALL CARTN                                                         ASDF3638
      CALL ANMODE                                                        ASDF3639
      WRITE (6,103)                                                      ASDF3640
103   FORMAT (22X,'>')                                                   ASDF3641
      READ (5,104) X,Y                                                   ASDF3642
104   FORMAT(F10.7,5X,F10.7)                                             ASDF3643
      CALL RECOVR                                                        ASDF3644
      IF(SQRT(X**2+Y**2).GT.1.0)IERROR=10                                ASDF3645
      IF(IERROR.NE.0)RETURN                                              ASDF3646
      CALL NEWLIN                                                        ASDF3647
      CALL NEWLIN                                                        ASDF3648
      CALL ANMODE                                                        ASDF3649
      Z=ABS(Y)                                                           ASDF3650
      WRITE(6,105) X,Z                                                   ASDF3651
105   FORMAT(9X,29HTHE NEW COMPLEX POLE WILL BE ,F10.7,6H +/- J,F10.7)   ASDF3652
      CALL RECOVR                                                        ASDF3653
      CALL NEWLIN                                                        ASDF3654
      CALL ANMODE                                                        ASDF3655
      WRITE(6,106)                                                       ASDF3656
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                    ASDF3657
      CALL RECOVR                                                        ASDF3658
      CALL SCURSR(ICHAR,IX,IY)                                           ASDF3659
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                           ASDF3660
      CALL NUC                                                           ASDF3661
      CALL STRRTS(4,1,X,Y)                                               ASDF3662
      RETURN                                                             ASDF3663
      END                                                                ASDF3664
                                                                         ASDF3665
      SUBROUTINE ARZRCT                                                  ASDF3666
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                               ASDF3667
      DIMENSION LTRARZ(64)                                               ASDF3668
      DATA LTRARZ/4HTO A,4HDD A,4H REA,4HL ZE,4HRO T,4HO TH,4HE DI       ASDF3669
     X    ,4HS-  ,4HPLAY,4H ENT,4HER T,4HHE R,4HEAL ,4HPART,4H WIT       ASDF3670
     X    ,4HHIN ,4HTHE ,4HBOX ,4HPROV,4HIDED,4H. TH,4HE NU,4HMBER       ASDF3671
     X    ,4H IS ,4HTO I,4HNCLU,4HDE A,4H DEC,4HIMAL,4H AND,4H MIN       ASDF3672
```

```
     X     ,4HUS   ,4HSIGN,4H AS ,4HAPPR,4HOPRI,4HATE.,4H  TH,4HE BO      ASDF3673
     X     ,4HX    ,4HMUST,4H CON,4HTAIN,4H SIG,4HN, D,4HECIM,4HAL A      ASDF3674
     X     ,4HND   ,4HNUMB,4HER. ,4H NOT,4HE: T,4HHE M,4HAGNI,4HTUDE      ASDF3675
     X     ,4H OF  ,4HTHE ,4HZERO,4H MUS,4HT BE,4H LES,4HS TH,4HEN T/     ASDF3676
      DATA LTRARZ(64)/4HEN. /                                             ASDF3677
      CALL CHCKSZ(1,1)                                                    ASDF3678
      IF(IERROR.NE.0)RETURN                                               ASDF3679
 10   CALL ERASE                                                          ASDF3680
      CALL MOVABS(0,528)                                                  ASDF3681
      DO 30 I=1,57,8                                                      ASDF3682
      CALL ANMODE                                                         ASDF3683
      WRITE(6,100)(LTRARZ(I+J-1),J=1,8)                                   ASDF3684
      CALL RECOVR                                                         ASDF3685
 30   CALL NEWLIN                                                         ASDF3686
 100  FORMAT (21X,8A4)                                                    ASDF3687
      CALL NEWLIN                                                         ASDF3688
      CALL ANMODE                                                         ASDF3689
      WRITE (6,101)                                                       ASDF3690
 101  FORMAT (27X,4HREAL)                                                 ASDF3691
      CALL RECOVR                                                         ASDF3692
      CALL NEWLIN                                                         ASDF3693
      CALL MOVABS(320,302)                                                ASDF3694
      CALL DRWREL(0,22)                                                   ASDF3695
      CALL DRWREL(138,0)                                                  ASDF3696
      CALL DRWREL(0,-22)                                                  ASDF3697
      CALL DRWREL(-138,0)                                                 ASDF3698
      CALL MOVREL(0,3)                                                    ASDF3699
      CALL CARTN                                                          ASDF3700
      CALL ANMODE                                                         ASDF3701
      WRITE (6,103)                                                       ASDF3702
 103  FORMAT (22X,'>')                                                    ASDF3703
      READ (5,104) X                                                      ASDF3704
 104  FORMAT(F10.7)                                                       ASDF3705
      CALL RECOVR                                                         ASDF3706
      IF(ABS(X).GE.10.)IERROR=12                                          ASDF3707
      IF(IERROR.NE.0)RETURN                                               ASDF3708
      CALL NEWLIN                                                         ASDF3709
      CALL NEWLIN                                                         ASDF3710
      CALL ANMODE                                                         ASDF3711
      Z=0.                                                                ASDF3712
      WRITE(6,105) X,Z                                                    ASDF3713
 105  FORMAT(15X,26HTHE NEW REAL ZERO WILL BE ,F10.7,3H +J,F10.7)         ASDF3714
      CALL RECOVR                                                         ASDF3715
      CALL NEWLIN                                                         ASDF3716
      CALL ANMODE                                                         ASDF3717
      WRITE(6,106)                                                        ASDF3718
 106  FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                     ASDF3719
      CALL RECOVR                                                         ASDF3720
```

```
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF3721
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF3722
      CALL NUC                                                          ASDF3723
      CALL STRRTS(1,1,X,0.)                                             ASDF3724
      RETURN                                                            ASDF3725
      END                                                               ASDF3726
                                                                        ASDF3727
      SUBROUTINE ARPRCT                                                 ASDF3728
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF3729
      DIMENSION LTRARP(64)                                              ASDF3730
      DATA LTRARP/4HTO A,4HDD A,4H REA,4HL PO,4HLE T,4HO TH,4HE DI      ASDF3731
     X     ,4HS-  ,4HPLAY,4H ENT,4HER T,4HHE R,4HEAL ,4HPART,4H WIT     ASDF3732
     X     ,4HHIN ,4HTHE ,4HBOX ,4HPROV,4HIDED,4H. TH,4HE NU,4HMBER     ASDF3733
     X     ,4H IS ,4HTO I,4HNCLU,4HDE A,4H DEC,4HIMAL,4H AND,4H MIN     ASDF3734
     X     ,4HUS  ,4HSIGN,4H AS ,4HAPPR,4HOPRI,4HATE.,4H  TH,4HE BO     ASDF3735
     X     ,4HX   ,4HMUST,4H CON,4HTAIN,4H SIG,4HN, D,4HECIM,4HAL A     ASDF3736
     X     ,4HND  ,4HNUMB,4HER. ,4H NOT,4HE: T,4HHE M,4HAGNI,4HTUDE     ASDF3737
     X     ,4H OF ,4HTHE ,4HPOLE,4H MUS,4HT BE,4H ONE,4H OR ,4HLESS/    ASDF3738
      DATA LTRARP(64)/4H.   /                                           ASDF3739
      CALL CHCKSZ(1,3)                                                  ASDF3740
      IF(IERROR.NE.0)RETURN                                             ASDF3741
   10 CALL ERASE                                                        ASDF3742
      CALL MOVABS(0,528)                                                ASDF3743
      DO 30 I=1,57,8                                                    ASDF3744
      CALL ANMODE                                                       ASDF3745
      WRITE(6,100)(LTRARP(I+J-1),J=1,8)                                 ASDF3746
      CALL RECOVR                                                       ASDF3747
   30 CALL NEWLIN                                                       ASDF3748
  100 FORMAT (21X,8A4)                                                  ASDF3749
      CALL NEWLIN                                                       ASDF3750
      CALL ANMODE                                                       ASDF3751
      WRITE (6,101)                                                     ASDF3752
  101 FORMAT (27X,4HREAL)                                               ASDF3753
      CALL RECOVR                                                       ASDF3754
      CALL NEWLIN                                                       ASDF3755
      CALL MOVABS(320,302)                                              ASDF3756
      CALL DRWREL(0,22)                                                 ASDF3757
      CALL DRWREL(138,0)                                                ASDF3758
      CALL DRWREL(0,-22)                                                ASDF3759
      CALL DRWREL(-138,0)                                               ASDF3760
      CALL MOVREL(0,3)                                                  ASDF3761
      CALL CARTN                                                        ASDF3762
      CALL ANMODE                                                       ASDF3763
      WRITE (6,103)                                                     ASDF3764
  103 FORMAT (22X,'>')                                                  ASDF3765
      READ (5,104) X                                                    ASDF3766
  104 FORMAT(F10.7)                                                     ASDF3767
      CALL RECOVR                                                       ASDF3768
```

```
      IF(ABS(X).GT.1.0)IERROR=13                                        ASDF3769
      IF(IERROR.NE.0)CALL ERROR                                         ASDF3770
      IF(IERROR.NE.0)RETURN                                             ASDF3771
      CALL NEWLIN                                                       ASDF3772
      CALL NEWLIN                                                       ASDF3773
      CALL ANMODE                                                       ASDF3774
      Z=0.                                                              ASDF3775
      WRITE(6,105) X,Z                                                  ASDF3776
105   FORMAT(15X,26HTHE NEW REAL POLE WILL BE ,F10.7,3H +J,F10.7)       ASDF3777
      CALL RECOVR                                                       ASDF3778
      CALL NEWLIN                                                       ASDF3779
      CALL ANMODE                                                       ASDF3780
      WRITE(6,106)                                                      ASDF3781
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF3782
      CALL RECOVR                                                       ASDF3783
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF3784
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF3785
      CALL NUC                                                          ASDF3786
      CALL STRRTS(3,1,X,0.)                                             ASDF3787
      RETURN                                                            ASDF3788
      END                                                               ASDF3789
                                                                        ASDF3790
      SUBROUTINE DCZRCT                                                 ASDF3791
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF3792
      DIMENSION LTRDCZ(64)                                              ASDF3793
      DATA LTRDCZ/4HTO D,4HELET,4HE A ,4HCOMP,4HLEX ,4HZERO,4H FRO      ASDF3794
     X     ,4HM   ,4HTHE ,4HDISP,4HLAY,,4H ENT,4HER T,4HHE ',4HX' A     ASDF3795
     X     ,4HNO  ,4H'Y' ,4HCOOR,4HDINA,4HTES ,4HOF A,4H POI,4HNT N     ASDF3796
     X     ,4HEAR ,4HONE ,4HOF T,4HHE R,4HOOTS,4H TO ,4HBE D,4HELET     ASDF3797
     X     ,4HED. ,4HENTE,4HR TH,4HESE ,4HVALU,4HES W,4HITHI,4HN TH     ASDF3798
     X     ,4HE   ,4HBOXE,4HS. I,4HNCLU,4HDE A,4H DEC,4HIMAL,4H, AN     ASDF3799
     X     ,4HD   ,4HMINU,4HS SI,4HGN A,4HS AP,4HPROP,4HRIAT,4HE.       ASDF3800
     X     ,4HNO  ,4HSIGN,4H IND,4HICAT,4HES P,4HOSIT,4HIVE ,4HVALU/    ASDF3801
      DATA LTRDCZ(64)/4HES. /                                           ASDF3802
      CALL CHCKSZ(-1,2)                                                 ASDF3803
      IF(IERROR.NE.0)RETURN                                             ASDF3804
10    CALL ERASE                                                        ASDF3805
      CALL MOVABS(0,528)                                                ASDF3806
      DO 30 I=1,57,8                                                    ASDF3807
      CALL ANMODE                                                       ASDF3808
      WRITE(6,100)(LTRDCZ(I+J-1),J=1,8)                                 ASDF3809
      CALL RECOVR                                                       ASDF3810
30    CALL NEWLIN                                                       ASDF3811
100   FORMAT (21X,8A4)                                                  ASDF3812
      CALL NEWLIN                                                       ASDF3813
      CALL ANMODE                                                       ASDF3814
      WRITE (6,101)                                                     ASDF3815
101   FORMAT(29X,'X',12X,'Y')                                           ASDF3816
```

```
      CALL RECOVR                                                       ASDF3817
      CALL NEWLIN                                                       ASDF3818
      CALL MOVABS(320,302)                                              ASDF3819
      CALL DRWREL(0,22)                                                 ASDF3820
      CALL DRWREL(138,0)                                                ASDF3821
      CALL DRWREL(0,-22)                                                ASDF3822
      CALL DRWREL(-138,0)                                               ASDF3823
      CALL MOVREL(184,0)                                                ASDF3824
      CALL DRWREL(0,22)                                                 ASDF3825
      CALL DRWREL(140,0)                                                ASDF3826
      CALL DRWREL(0,-22)                                                ASDF3827
      CALL DRWREL(-140,0)                                               ASDF3828
      CALL MOVREL(0,3)                                                  ASDF3829
      CALL CARTN                                                        ASDF3830
      CALL ANMODE                                                       ASDF3831
      WRITE (6,103)                                                     ASDF3832
  103 FORMAT (22X,'>')                                                  ASDF3833
      READ (5,104) X,Y                                                  ASDF3834
  104 FORMAT(F10.7,5X,F10.7)                                            ASDF3835
      CALL RECOVR                                                       ASDF3836
      CALL LOCATE(2,X,Y,LINEPZ)                                         ASDF3837
      CALL NEWLIN                                                       ASDF3838
      CALL NEWLIN                                                       ASDF3839
      CALL ANMODE                                                       ASDF3840
      WRITE(6,105)POLZRO(LINEPZ,1),POLZRO(LINEPZ,2)                     ASDF3841
  105 FORMAT(11X,26HDELETED ZERO PAIR WILL BE ,F10.7,6H +/- J,F10.7)    ASDF3842
      CALL RECOVR                                                       ASDF3843
      CALL NEWLIN                                                       ASDF3844
      CALL ANMODE                                                       ASDF3845
      WRITE(6,106)                                                      ASDF3846
  106 FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF3847
      CALL RECOVR                                                       ASDF3848
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF3849
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF3850
      CALL UPDATE(2,LINEPZ)                                             ASDF3851
      CALL NUC                                                          ASDF3852
      RETURN                                                            ASDF3853
      END                                                               ASDF3854
                                                                        ASDF3855
      SUBROUTINE DCPRCT                                                 ASDF3856
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF3857
      DIMENSION LTRDCP(64)                                              ASDF3858
      DATA LTRDCP/4HTO D,4HELET,4HE A ,4HCOMP,4HLEX ,4HPOLE,4H FRO      ASDF3859
     X    ,4HM   ,4HTHE ,4HUNIT,4H CIR,4HCLE,,4H ENT,4HER T,4HHE '      ASDF3860
     X    ,4HX'  ,4HAND ,4H'Y' ,4HCOOR,4HDINA,4HTES ,4HOF T,4HHE P      ASDF3861
     X    ,4HOINT,4HNEAR,4H ONE,4H OF ,4HTHE ,4HROOT,4HS TO,4H BE       ASDF3862
     X    ,4HDEL-,4HETED,4H. EN,4HTER ,4HTHE ,4HVALU,4HES I,4HN TH      ASDF3863
     X    ,4HE   ,4HBOXE,4HS. I,4HNCLU,4HDE A,4H DEC,4HIMAL,4H, AN      ASDF3864
```

```
     X    ,4HD   ,4HMINU,4HS SI,4HGN A,4HS AP,4HPROP,4HRIAT,4HE.        ASDF3865
     X    ,4HNO  ,4HSIGN,4H IND,4HICAT,4HES P,4HOSIT,4HIVE ,4HVALU/     ASDF3866
      DATA LTRDCP(64)/4HES. /                                           ASDF3867
      CALL CHCKSZ(-1,4)                                                 ASDF3868
      IF(IERROR.NE.0)RETURN                                             ASDF3869
 10   CALL ERASE                                                        ASDF3870
      CALL MOVABS(0,528)                                                ASDF3871
      DO 30 I=1,57,8                                                    ASDF3872
      CALL ANMODE                                                       ASDF3873
      WRITE(6,100)(LTRDCP(I+J-1),J=1,8)                                 ASDF3874
      CALL RECOVR                                                       ASDF3875
 30   CALL NEWLIN                                                       ASDF3876
      CALL NEWLIN                                                       ASDF3877
      CALL ANMODE                                                       ASDF3878
      WRITE (6,101)                                                     ASDF3879
 100  FORMAT (21X,8A4)                                                  ASDF3880
 101  FORMAT(29X,'X',12X,'Y')                                           ASDF3881
      CALL RECOVR                                                       ASDF3882
      CALL NEWLIN                                                       ASDF3883
      CALL MOVABS(320,302)                                              ASDF3884
      CALL DRWREL(0,22)                                                 ASDF3885
      CALL DRWREL(138,0)                                                ASDF3886
      CALL DRWREL(0,-22)                                                ASDF3887
      CALL DRWREL(-138,0)                                               ASDF3888
      CALL MOVREL(184,0)                                                ASDF3889
      CALL DRWREL(0,22)                                                 ASDF3890
      CALL DRWREL(140,0)                                                ASDF3891
      CALL DRWREL(0,-22)                                                ASDF3892
      CALL DRWREL(-140,0)                                               ASDF3893
      CALL MOVREL(0,3)                                                  ASDF3894
      CALL CARTN                                                        ASDF3895
      CALL ANMODE                                                       ASDF3896
      WRITE (6,103)                                                     ASDF3897
 103  FORMAT (22X,'>')                                                  ASDF3898
      READ (5,104) X,Y                                                  ASDF3899
 104  FORMAT(F10.7,5X,F10.7)                                            ASDF3900
      CALL RECOVR                                                       ASDF3901
      CALL LOCATE(4,X,Y,LINEPZ)                                         ASDF3902
      CALL NEWLIN                                                       ASDF3903
      CALL NEWLIN                                                       ASDF3904
      CALL ANMODE                                                       ASDF3905
      WRITE(6,105)POLZRO(LINEPZ,1),POLZRO(LINEPZ,2)                     ASDF3906
 105  FORMAT(11X,26HDELETED POLE PAIR WILL BE ,F10.7,6H +/- J,F10.7)    ASDF3907
      CALL RECOVR                                                       ASDF3908
      CALL LOCATE(4,X,Y,LINEPZ)                                         ASDF3909
      CALL NEWLIN                                                       ASDF3910
      CALL ANMODE                                                       ASDF3911
      WRITE(6,106)                                                      ASDF3912
```

```
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)
      CALL RECOVR
      CALL SCURSR(ICHAR,IX,IY)
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10
      CALL UPDATE(4,LINEPZ)
      CALL NUC
      RETURN
      END


      SUBROUTINE DRZRCT
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      DIMENSION LTRDRZ(64)
      DATA LTRDRZ/4HTO D,4HELET,4HE A ,4HREAL,4H ZER,4HO FR,4HOM T
     X    ,4HHE  ,4HDISP,4HLAY,,4H ENT,4HER T,4HHE ',4HX' C,4HOORD
     X    ,4HIN- ,4HATE ,4HOF A,4H POI,4HNT N,4HEAR ,4HTHE ,4HROOT
     X    ,4H TO ,4HBE D,4HELET,4HED. ,4H ENT,4HER T,4HHE V,4HALUE
     X    ,4H    ,4HWITH,4HIN T,4HHE B,4HOX P,4HROVI,4HDED.,4H INC
     X    ,4HLUDE,4HA DE,4HCIMA,4HL, A,4HND M,4HINUS,4H SIG,4HN AS
     X    ,4H    ,4HAPPR,4HOPRI,4HATE.,4H  NO,4H SIG,4HN IN,4HDICA
     X    ,4HTES ,4HA PO,4HSITI,4HVE V,4HALUE,4H.   ,4H    ,4H    /
      DATA LTRDRZ(64)/4H    /
      CALL CHCKSZ(-1,1)
      IF(IERROR.NE.0)RETURN
10    CALL ERASE
      CALL MOVABS(0,528)
      DO 30 I=1,57,8
      CALL ANMODE
      WRITE(6,100)(LTRDRZ(I+J-1),J=1,8)
      CALL RECOVR
30    CALL NEWLIN
100   FORMAT (21X,8A4)
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,101)
101   FORMAT(28X,'X')
      CALL RECOVR
      CALL NEWLIN
      CALL MOVABS(320,302)
      CALL DRWREL(0,22)
      CALL DRWREL(138,0)
      CALL DRWREL(0,-22)
      CALL DRWREL(-138,0)
      CALL MOVREL(0,3)
      CALL CARTN
      CALL ANMODE
      WRITE (6,103)
103   FORMAT (22X,'>')
```

```
      READ (5,104) X
104   FORMAT(F10.7)
      CALL RECOVR
      CALL LOCATE(1,X,0.,LINEPZ)
      CALL NEWLIN
      CALL NEWLIN
      CALL ANMODE
      Z=0.
      WRITE(6,105)POLZRO(LINEPZ,1),POLZRO(LINEPZ,2)
105   FORMAT(15X,26HDELETED REAL ZERO WILL BE ,F10.7,3H +J,F10.7)
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,106)
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)
      CALL RECOVR
      CALL SCURSR(ICHAR,IX,IY)
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10
      CALL UPDATE(1,LINEPZ)
      CALL NUC
      RETURN
      END

      SUBROUTINE DRPRCT
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      DIMENSION LTRDRP(64)
      DATA LTRDRP/4HTO D,4HELET,4HE A ,4HREAL,4H POL,4HE FR,4HOM T
     X     ,4HHE  ,4HUNIT,4H CIR,4HCLE.,4H ENT,4HER T,4HHE ',4HX' C
     X     ,4HOOR-,4HDINA,4HTE O,4HF A ,4HPOIN,4HT NE,4HAR T,4HHE R
     X     ,4HOOT ,4HTO B,4HE DE,4HLETE,4HD. E,4HNTER,4H THE,4H VAL
     X     ,4HUE  ,4HWITH,4HIN T,4HHE B,4HOX P,4HROVI,4HDED.,4H INC
     X     ,4HLUDE,4HA DE,4HCIMA,4HL, A,4HND M,4HINUS,4H SIG,4HN AS
     X     ,4H    ,4HAPPR,4HOPRI,4HATE.,4H  NO,4H SIG,4HN IN,4HDICA
     X     ,4HTES ,4HA PO,4HSITI,4HVE V,4HALUE,4H.   ,4H    ,4H    /
      DATA LTRDRP(64)/4H    /
      CALL CHCKSZ(-1,3)
      IF(IERROR.NE.0)RETURN
10    CALL ERASE
      CALL MOVABS(0,528)
      DO 30 I=1,57,8
      CALL ANMODE
      WRITE(6,100)(LTRDRP(I+J-1),J=1,8)
      CALL RECOVR
30    CALL NEWLIN
100   FORMAT (21X,8A4)
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,101)
```

```
101   FORMAT(28X,'X')                                                    ASDF4009
      CALL RECOVR                                                        ASDF4010
      CALL NEWLIN                                                        ASDF4011
      CALL MOVABS(320,302)                                               ASDF4012
      CALL DRWREL(0,22)                                                  ASDF4013
      CALL DRWREL(138,0)                                                 ASDF4014
      CALL DRWREL(0,-22)                                                 ASDF4015
      CALL DRWREL(-138,0)                                                ASDF4016
      CALL MOVREL(0,3)                                                   ASDF4017
      CALL CARTN                                                         ASDF4018
      CALL ANMODE                                                        ASDF4019
      WRITE (6,103)                                                      ASDF4020
103   FORMAT (22X,'>')                                                   ASDF4021
      READ (5,104) X                                                     ASDF4022
104   FORMAT(F10.7)                                                      ASDF4023
      CALL RECOVR                                                        ASDF4024
      CALL LOCATE(3,X,0.,LINEPZ)                                         ASDF4025
      CALL NEWLIN                                                        ASDF4026
      CALL NEWLIN                                                        ASDF4027
      CALL ANMODE                                                        ASDF4028
      WRITE(6,105)POLZRO(LINEPZ,1),POLZRO(LINEPZ,2)                      ASDF4029
105   FORMAT(15X,26HDELETED REAL POLE WILL BE ,F10.7,3H +J,F10.7)        ASDF4030
      CALL RECOVR                                                        ASDF4031
      CALL NEWLIN                                                        ASDF4032
      CALL ANMODE                                                        ASDF4033
      WRITE(6,106)                                                       ASDF4034
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                    ASDF4035
      CALL RECOVR                                                        ASDF4036
      CALL SCURSR(ICHAR,IX,IY)                                           ASDF4037
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                           ASDF4038
      CALL UPDATE(3,LINEPZ)                                              ASDF4039
      CALL NUC                                                           ASDF4040
      RETURN                                                             ASDF4041
      END                                                                ASDF4042
                                                                         ASDF4043
      SUBROUTINE ACZPLR                                                  ASDF4044
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                               ASDF4045
      DIMENSION LTPACZ(64)                                               ASDF4046
      DATA LTPACZ/4HTO A,4HDD A,4H COM,4HPLEX,4H ZER,4HO TO,4H THE       ASDF4047
     X     ,4H    ,4HDISP,4HLAY,,4H ENT,4HER T,4HHE R,4HADIA,4HL AN      ASDF4048
     X     ,4HD   ,4HTHET,4HA CO,4HMPON,4HENTS,4H WIT,4HHIN ,4HPROV      ASDF4049
     X     ,4HIDED,4HBOXE,4HS. T,4HHE M,4HAGNI,4HTUDE,4H OF ,4HRHO       ASDF4050
     X     ,4HMUST,4HBE L,4HESS ,4HTHAN,4H TEN,4H, TH,4HETA(,4HRADI      ASDF4051
     X     ,4HANS),4HEACH,4H NUM,4HBER ,4HREQU,4HIRES,4H A D,4HECIM      ASDF4052
     X     ,4HAL, ,4HAND ,4HAS A,4HPPRO,4HPRIA,4HTE A,4H MIN,4HUS S      ASDF4053
     X     ,4HIGN.,4HBOTH,4H ARE,4H CON,4HFINE,4HD TO,4H THE,4H BOX/     ASDF4054
      DATA LTPACZ(64)/4HES. /                                            ASDF4055
      CALL CHCKSZ(2,2)                                                   ASDF4056
```

```
      IF(IERROR.NE.0)RETURN                                             ASDF4057
10    CALL ERASE                                                        ASDF4058
      CALL MOVABS(0,528)                                                ASDF4059
      DO 30 I=1,57,8                                                    ASDF4060
      CALL ANMODE                                                       ASDF4061
      WRITE(6,100)(LTPACZ(I+J-1),J=1,8)                                 ASDF4062
      CALL RECOVR                                                       ASDF4063
30    CALL NEWLIN                                                       ASDF4064
100   FORMAT (21X,8A4)                                                  ASDF4065
      CALL NEWLIN                                                       ASDF4066
      CALL ANMODE                                                       ASDF4067
      WRITE (6,101)                                                     ASDF4068
101   FORMAT(25X,'MAGNITUDE',8X,'ANGLE')                                ASDF4069
      CALL RECOVR                                                       ASDF4070
      CALL NEWLIN                                                       ASDF4071
      CALL ANMODE                                                       ASDF4072
      WRITE (6,102)                                                     ASDF4073
102   FORMAT (34X,3H+/-)                                                ASDF4074
      CALL RECOVR                                                       ASDF4075
      CALL ANGLE(504,304)                                               ASDF4076
      CALL MOVABS(320,302)                                              ASDF4077
      CALL DRWREL(0,22)                                                 ASDF4078
      CALL DRWREL(138,0)                                                ASDF4079
      CALL DRWREL(0,-22)                                                ASDF4080
      CALL DRWREL(-138,0)                                               ASDF4081
      CALL MOVREL(212,0)                                                ASDF4082
      CALL DRWREL(0,22)                                                 ASDF4083
      CALL DRWREL(140,0)                                                ASDF4084
      CALL DRWREL(0,-22)                                                ASDF4085
      CALL DRWREL(-140,0)                                               ASDF4086
      CALL MOVREL(0,3)                                                  ASDF4087
      CALL CARTN                                                        ASDF4088
      CALL ANMODE                                                       ASDF4089
      WRITE (6,103)                                                     ASDF4090
103   FORMAT (22X,'>')                                                  ASDF4091
      READ (5,104) RHO,THETA                                            ASDF4092
104   FORMAT(F10.7,5X,F10.7)                                            ASDF4093
      CALL RECOVR                                                       ASDF4094
      IF(RHO.GE.10)IERROR=14                                            ASDF4095
      IF(IERROR.NE.0)CALL ERROR                                         ASDF4096
      IF(IERROR.NE.0)RETURN                                             ASDF4097
      CALL NEWLIN                                                       ASDF4098
      CALL NEWLIN                                                       ASDF4099
      CALL ANMODE                                                       ASDF4100
      THETA=ABS(THETA)                                                  ASDF4101
      WRITE(6,105) RHO,THETA                                            ASDF4102
105   FORMAT(9X,30HTHE NEW COMPLEX ZEROS WILL BE ,F10.7,6H +/-  ,F10.7) ASDF4103
      CALL RECOVR                                                       ASDF4104
```

```
      CALL ANGLE(727,260)                                               ASDF4105
      CALL NEWLIN                                                       ASDF4106
      CALL ANMODE                                                       ASDF4107
      WRITE(6,106)                                                      ASDF4108
  106 FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF4109
      CALL RECOVR                                                       ASDF4110
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF4111
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF4112
      CALL NUC                                                          ASDF4113
      CALL STRRTS(2,2,RHO,THETA)                                        ASDF4114
      RETURN                                                            ASDF4115
      END                                                               ASDF4116
                                                                        ASDF4117
      SUBROUTINE ACPPLR                                                 ASDF4118
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF4119
      DIMENSION LTPACP(64)                                              ASDF4120
      DATA LTPACP/4HTO A,4HDD A,4H COM,4HPLEX,4H POL,4HE WI,4HTHIN      ASDF4121
     X    ,4H THE,4HUNIT,4H CIR,4HCLE ,4HENTE,4HR TH,4HE RA,4HDIAL      ASDF4122
     X    ,4H AND,4HTHET,4HA CO,4HMPON,4HENTS,4H WIT,4HHIN ,4HPROV      ASDF4123
     X    ,4HIDED,4HBOXE,4HS.  ,4HRHO ,4HMUST,4H BE ,4HONE ,4HOR L      ASDF4124
     X    ,4HESS.,4HTHET,4HA MU,4HST B,4HE EN,4HTERE,4HD IN,4H RAD      ASDF4125
     X    ,4HIANS,4HEACH,4H NUM,4HBER ,4HREQU,4HIRES,4H A D,4HECIM      ASDF4126
     X    ,4HAL, ,4HAND ,4HAS A,4HPPRO,4HPRIA,4HTE A,4H MIN,4HUS S      ASDF4127
     X    ,4HIGN.,4HBOTH,4H ARE,4H CON,4HFINE,4HD TO,4H THE,4H BOX/     ASDF4128
      DATA LTPACP(64)/4HES. /                                           ASDF4129
      CALL CHCKSZ(2,4)                                                  ASDF4130
      IF(IERROR.NE.0)RETURN                                             ASDF4131
   10 CALL ERASE                                                        ASDF4132
      CALL MOVABS(0,528)                                                ASDF4133
      DO 30 I=1,57,8                                                    ASDF4134
      CALL ANMODE                                                       ASDF4135
      WRITE(6,100)(LTPACP(I+J-1),J=1,8)                                 ASDF4136
      CALL RECOVR                                                       ASDF4137
   30 CALL NEWLIN                                                       ASDF4138
  100 FORMAT (21X,8A4)                                                  ASDF4139
      CALL NEWLIN                                                       ASDF4140
      CALL ANMODE                                                       ASDF4141
      WRITE (6,101)                                                     ASDF4142
  101 FORMAT(25X,'MAGNITUDE',8X,'ANGLE')                                ASDF4143
      CALL RECOVR                                                       ASDF4144
      CALL NEWLIN                                                       ASDF4145
      CALL ANMODE                                                       ASDF4146
      WRITE (6,102)                                                     ASDF4147
  102 FORMAT (34X,3H+/-)                                                ASDF4148
      CALL RECOVR                                                       ASDF4149
      CALL ANGLE(504,304)                                               ASDF4150
      CALL MOVABS(320,302)                                              ASDF4151
      CALL DRWREL(0,22)                                                 ASDF4152
```

```
      CALL DRWREL(138,0)                                                ASDF4153
      CALL DRWREL(0,-22)                                                ASDF4154
      CALL DRWREL(-138,0)                                               ASDF4155
      CALL MOVREL(212,0)                                                ASDF4156
      CALL DRWREL(0,22)                                                 ASDF4157
      CALL DRWREL(140,0)                                                ASDF4158
      CALL DRWREL(0,-22)                                                ASDF4159
      CALL DRWREL(-140,0)                                               ASDF4160
      CALL MOVREL(0,3)                                                  ASDF4161
      CALL CARTN                                                        ASDF4162
      CALL ANMODE                                                       ASDF4163
      WRITE (6,103)                                                     ASDF4164
103   FORMAT (22X,'>')                                                  ASDF4165
      READ (5,104) RHO,THETA                                            ASDF4166
104   FORMAT(F10.7,5X,F10.7)                                            ASDF4167
      CALL RECOVR                                                       ASDF4168
      IF(RHO.GT.1.0)IERROR=15                                           ASDF4169
      IF(IERROR.NE.0)RETURN                                             ASDF4170
      CALL NEWLIN                                                       ASDF4171
      CALL NEWLIN                                                       ASDF4172
      CALL ANMODE                                                       ASDF4173
      THETA=ABS(THETA)                                                  ASDF4174
      WRITE(6,105) RHO,THETA                                            ASDF4175
105   FORMAT(9X,30HTHE NEW COMPLEX POLES WILL BE ,F10.7,6H +/-  ,F10.7) ASDF4176
      CALL RECOVR                                                       ASDF4177
      CALL ANGLE(727,260)                                               ASDF4178
      CALL NEWLIN                                                       ASDF4179
      CALL ANMODE                                                       ASDF4180
      WRITE(6,106)                                                      ASDF4181
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF4182
      CALL RECOVR                                                       ASDF4183
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF4184
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF4185
      CALL NUC                                                          ASDF4186
      CALL STRRTS(4,2,RHO,THETA)                                        ASDF4187
      RETURN                                                            ASDF4188
      END                                                               ASDF4189
                                                                        ASDF4190
      SUBROUTINE ARZPLR                                                 ASDF4191
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF4192
      DIMENSION LTPARZ(64)                                              ASDF4193
      DATA LTPARZ/4HTO A,4HDD A,4H REA,4HL ZE,4HRO T,4HO TH,4HE DI      ASDF4194
     X    ,4HS-  ,4HPLAY,4H, EN,4HTER ,4HTHE ,4HVALU,4HE OF,4H RHO      ASDF4195
     X    ,4H IN ,4HTHE ,4HBOX ,4HPROV,4HIDED,4H. TH,4HE MA,4HGNIT      ASDF4196
     X    ,4HUDE ,4HOF R,4HHO M,4HUST ,4HBE L,4HESS ,4HTHAN,4H TEN      ASDF4197
     X    ,4H.THE,4HNUMB,4HER, ,4HINCL,4HUDIN,4HG DE,4HCIMA,4HL AN      ASDF4198
     X    ,4HD   ,4HSIGN,4H AS ,4HAPPR,4HOPRI,4HATE ,4HMUST,4H BE       ASDF4199
     X    ,4HCON-,4HTAIN,4HED W,4HITHI,4HN TH,4HE BO,4HX.  ,4HNO S      ASDF4200
```

```
     X    ,4HIGN ,4HINDI,4HCATE,4HS A ,4HPOSI,4HTIVE,4H VAL,4HUE. /
      DATA LTPARZ(64)/4H    /
      CALL CHCKSZ(1,1)
      IF(IERROR.NE.0)RETURN
10    CALL ERASE
      CALL MOVABS(0,528)
      DO 30 I=1,57,8
      CALL ANMODE
      WRITE(6,100)(LTPARZ(I+J-1),J=1,8)
      CALL RECOVR
30    CALL NEWLIN
100   FORMAT (21X,8A4)
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,101)
101   FORMAT(25X,'MAGNITUDE')
      CALL RECOVR
      CALL NEWLIN
      CALL MOVABS(320,302)
      CALL DRWREL(0,22)
      CALL DRWREL(138,0)
      CALL DRWREL(0,-22)
      CALL DRWREL(-138,0)
      CALL MOVREL(0,3)
      CALL CARTN
      CALL ANMODE
      WRITE (6,103)
103   FORMAT (22X,'>')
      READ (5,104) RHO
104   FORMAT(F10.7)
      CALL RECOVR
      IF(RHO.GT.10)IERROR=16
      IF(IERROR.NE.0)RETURN
      CALL NEWLIN
      CALL NEWLIN
      CALL ANMODE
      THETA=0.
      WRITE(6,105) RHO,THETA
105   FORMAT( 12X, 26HTHE NEW REAL ZERO WILL BE ,F10.7,3X,F10.7)
      CALL RECOVR
      CALL ANGLE(677,260)
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,106)
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)
      CALL RECOVR
      CALL SCURSR(ICHAR,IX,IY)
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10
```

```
      CALL NUC
      CALL STRRTS(1,2,RHO,0.)
      RETURN
      END

      SUBROUTINE ARPPLR
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      DIMENSION LTPARP(64)
      DATA LTPARP/4HTO A,4HDD A,4H REA,4HL PO,4HLE T,4HO TH,4HE DI
     X     ,4HS-  ,4HPLAY,4H ENT,4HER A,4H VAL,4HUE O,4HF RH,4HO, (
     X     ,4HONE ,4HOR L,4HESS),4H IN ,4HTHE ,4HBOX ,4HPROV,4HIDED
     X     ,4H.   ,4HTHE ,4HNUMB,4HER, ,4HINCL,4HUDIN,4HG DE,4HCIMA
     X     ,4HL   ,4HAND ,4HSIGN,4H AS ,4HAPPR,4HOPRI,4HATE ,4HMUST
     X     ,4H BE ,4HCONT,4HAINE,4HD WI,4HTHIN,4H THE,4H BOX,4H.
     X     ,4H    ,4HNOTE,4H: TH,4HE MA,4HGNIT,4HUDE ,4HOF R,4HHO M
     X     ,4HUST ,4HBE O,4HNE O,4HR LE,4HSS. ,4H    ,4H    ,4H    /
      DATA LTPARP(64)/4H    /
      CALL CHCKSZ(1,3)
      IF(IERROR.NE.0)RETURN
   10 CALL ERASE
      CALL MOVABS(0,528)
      DO 30 I=1,57,8
      CALL ANMODE
      WRITE(6,100)(LTPARP(I+J-1),J=1,8)
      CALL RECOVR
   30 CALL NEWLIN
  100 FORMAT (21X,8A4)
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,101)
  101 FORMAT(25X,'MAGNITUDE')
      CALL RECOVR
      CALL NEWLIN
      CALL MOVABS(320,302)
      CALL DRWREL(0,22)
      CALL DRWREL(138,0)
      CALL DRWREL(0,-22)
      CALL DRWREL(-138,0)
      CALL MOVREL(0,3)
      CALL CARTN
      CALL ANMODE
      WRITE (6,103)
  103 FORMAT (22X,'>')
      READ (5,104) RHO
  104 FORMAT(F10.7)
      CALL RECOVR
      IF(RHO.GT.1.0)IERROR=17
      IF(IERROR.NE.0)RETURN
```

```
      CALL NEWLIN                                                       ASDF4297
      CALL NEWLIN                                                       ASDF4298
      CALL ANMODE                                                       ASDF4299
      THETA=0.                                                          ASDF4300
      WRITE(6,105) RHO,THETA                                            ASDF4301
105   FORMAT(12X,26HTHE NEW REAL POLE WILL BE ,F10.7,3X,F10.7)          ASDF4302
      CALL RECOVR                                                       ASDF4303
      CALL ANGLE(677,260)                                               ASDF4304
      CALL NEWLIN                                                       ASDF4305
      CALL ANMODE                                                       ASDF4306
      WRITE(6,106)                                                      ASDF4307
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF4308
      CALL RECOVR                                                       ASDF4309
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF4310
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF4311
      CALL NUC                                                          ASDF4312
      CALL STRRTS(3,2,RHO,0.)                                           ASDF4313
      RETURN                                                            ASDF4314
      END                                                               ASDF4315
                                                                        ASDF4316
      SUBROUTINE DCZPLR                                                 ASDF4317
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF4318
      DIMENSION LTPDCZ(64)                                              ASDF4319
      DATA LTPDCZ/4HTO D,4HELET,4HE A ,4HCOMP,4HLEX ,4HZERO,4H FRO      ASDF4320
     X     ,4HM   ,4HTHE ,4HDISP,4HLAY,,4H ENT,4HER T,4HHE ',4HRHO'     ASDF4321
     X     ,4H AND,4H'THE,4HTA' ,4HCOOR,4HDINA,4HTES ,4HOF A,4H POI     ASDF4322
     X     ,4HNT  ,4HNEAR,4H ONE,4H OF ,4HTHE ,4HROOT,4HS TO,4H BE      ASDF4323
     X     ,4HDEL-,4HETED,4H. EN,4HTER ,4HTHES,4HE VA,4HLUES,4H IN      ASDF4324
     X     ,4HTHE ,4HBOXE,4HS.  ,4HINCL,4HUDE ,4HA DE,4HCIMA,4HL, A     ASDF4325
     X     ,4HND  ,4HMUNU,4HS SI,4HGN A,4HS AP,4HPROP,4HRIAT,4HE.       ASDF4326
     X     ,4HNO  ,4HSIGN,4H IND,4HICAT,4HES P,4HOSIT,4HIVE ,4HVALU/    ASDF4327
      DATA LTPDCZ(64)/4HES. /                                           ASDF4328
      CALL CHCKSZ(-1,2)                                                 ASDF4329
      IF(IERROR.NE.0)RETURN                                             ASDF4330
10    CALL ERASE                                                        ASDF4331
      CALL MOVABS(0,528)                                                ASDF4332
      DO 30 I=1,57,8                                                    ASDF4333
      CALL ANMODE                                                       ASDF4334
      WRITE(6,100)(LTPDCZ(I+J-1),J=1,8)                                 ASDF4335
      CALL RECOVR                                                       ASDF4336
30    CALL NEWLIN                                                       ASDF4337
100   FORMAT (21X,8A4)                                                  ASDF4338
      CALL NEWLIN                                                       ASDF4339
      CALL ANMODE                                                       ASDF4340
      WRITE (6,101)                                                     ASDF4341
101   FORMAT(27X,'RHO',12X,'THETA')                                     ASDF4342
      CALL RECOVR                                                       ASDF4343
      CALL NEWLIN                                                       ASDF4344
```

```
      CALL ANMODE                                                       ASDF4345
      WRITE (6,102)                                                     ASDF4346
  102 FORMAT (34X,3H+/-)                                                ASDF4347
      CALL ANGLE(504,304)                                               ASDF4348
      CALL RECOVR                                                       ASDF4349
      CALL MOVABS(320,302)                                              ASDF4350
      CALL DRWREL(0,22)                                                 ASDF4351
      CALL DRWREL(138,0)                                                ASDF4352
      CALL DRWREL(0,-22)                                                ASDF4353
      CALL DRWREL(-138,0)                                               ASDF4354
      CALL MOVREL(212,0)                                                ASDF4355
      CALL DRWREL(0,22)                                                 ASDF4356
      CALL DRWREL(140,0)                                                ASDF4357
      CALL DRWREL(0,-22)                                                ASDF4358
      CALL DRWREL(-140,0)                                               ASDF4359
      CALL MOVREL(0,3)                                                  ASDF4360
      CALL CARTN                                                        ASDF4361
      CALL ANMODE                                                       ASDF4362
      WRITE (6,103)                                                     ASDF4363
  103 FORMAT (22X,'>')                                                  ASDF4364
      READ (5,104) RHO,THETA                                            ASDF4365
  104 FORMAT(F10.7,5X,F10.7)                                            ASDF4366
      CALL RECOVR                                                       ASDF4367
      X=RHO*COS(THETA)                                                  ASDF4368
      Y=RHO*SIN(THETA)                                                  ASDF4369
      CALL LOCATE(2,X,Y,LINEPZ)                                         ASDF4370
      CALL NEWLIN                                                       ASDF4371
      CALL NEWLIN                                                       ASDF4372
      CALL ANMODE                                                       ASDF4373
      WRITE(6,105)POLZRO(LINEPZ,3),POLZRO(LINEPZ,4)                     ASDF4374
  105 FORMAT(9X,30HDELETED COMPLEX ZEROS WILL BE ,F10.7,6H +/-  ,F10.7) ASDF4375
      CALL RECOVR                                                       ASDF4376
      CALL ANGLE(727,260)                                               ASDF4377
      CALL NEWLIN                                                       ASDF4378
      CALL ANMODE                                                       ASDF4379
      WRITE(6,106)                                                      ASDF4380
  106 FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF4381
      CALL RECOVR                                                       ASDF4382
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF4383
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF4384
      CALL UPDATE(2,LINEPZ)                                             ASDF4385
      CALL NUC                                                          ASDF4386
      RETURN                                                            ASDF4387
      END                                                               ASDF4388
                                                                        ASDF4389
      SUBROUTINE DCPPLR                                                 ASDF4390
      COMMON IXXXXX(4),POLZRO(20,5),IERROR                              ASDF4391
      DIMENSION LTPDCP(64)                                              ASDF4392
```

```
      DATA LTPDCP/4HTO D,4HELET,4HE A ,4HCOMP,4HLEX ,4HPOLE,4H FRO       ASDF4393
     X    ,4HM   ,4HTHE ,4HUNIT,4H CIR,4HCLE,,4H ENT,4HER T,4HHE '       ASDF4394
     X    ,4HRHO',4HAND ,4H'THE,4HTA' ,4HCOOR,4HDINA,4HTES ,4HOF A       ASDF4395
     X    ,4H    ,4HPOIN,4HT NE,4HAR O,4HNE O,4HF TH,4HE RO,4HOTS        ASDF4396
     X    ,4HTO  ,4HBE D,4HELET,4HED. ,4H ENT,4HER T,4HHE V,4HALUE       ASDF4397
     X    ,4HS IN,4HTHE ,4HBOXE,4HS.  ,4HINCL,4HUDE ,4HA DE,4HCIMA       ASDF4398
     X    ,4HL,  ,4HAND ,4HMINU,4HS SI,4HGN A,4HS AP,4HPROP,4HRIAT       ASDF4399
     X    ,4HE.  ,4HNO S,4HIGN ,4HIMPL,4HIES ,4HPOSI,4HTIVE,4H VAL/      ASDF4400
      DATA LTPDCP(64)/4HUES./                                            ASDF4401
      CALL CHCKSZ(-1,4)                                                  ASDF4402
      IF(IERROR.NE.0)RETURN                                              ASDF4403
10    CALL ERASE                                                         ASDF4404
      CALL MOVABS(0,528)                                                 ASDF4405
      DO 30 I=1,57,8                                                     ASDF4406
      CALL ANMODE                                                        ASDF4407
      WRITE(6,100)(LTPDCP(I+J-1),J=1,8)                                  ASDF4408
      CALL RECOVR                                                        ASDF4409
30    CALL NEWLIN                                                        ASDF4410
100   FORMAT (21X,8A4)                                                   ASDF4411
      CALL NEWLIN                                                        ASDF4412
      CALL ANMODE                                                        ASDF4413
      WRITE (6,101)                                                      ASDF4414
101   FORMAT(27X,'RHO',12X,'THETA')                                      ASDF4415
      CALL RECOVR                                                        ASDF4416
      CALL NEWLIN                                                        ASDF4417
      CALL ANMODE                                                        ASDF4418
      WRITE (6,102)                                                      ASDF4419
102   FORMAT (34X,3H+/-)                                                 ASDF4420
      CALL RECOVR                                                        ASDF4421
      CALL ANGLE(504,304)                                                ASDF4422
      CALL MOVABS(320,302)                                               ASDF4423
      CALL DRWREL(0,22)                                                  ASDF4424
      CALL DRWREL(138,0)                                                 ASDF4425
      CALL DRWREL(0,-22)                                                 ASDF4426
      CALL DRWREL(-138,0)                                                ASDF4427
      CALL MOVREL(212,0)                                                 ASDF4428
      CALL DRWREL(0,22)                                                  ASDF4429
      CALL DRWREL(140,0)                                                 ASDF4430
      CALL DRWREL(0,-22)                                                 ASDF4431
      CALL DRWREL(-140,0)                                                ASDF4432
      CALL MOVREL(0,3)                                                   ASDF4433
      CALL CARTN                                                         ASDF4434
      CALL ANMODE                                                        ASDF4435
      WRITE (6,103)                                                      ASDF4436
103   FORMAT (22X,'>')                                                   ASDF4437
      READ (5,104) RHO,THETA                                             ASDF4438
104   FORMAT(F10.7,5X,F10.7)                                             ASDF4439
      CALL RECOVR                                                        ASDF4440
```

```
      X=RHO*COS(THETA)
      Y=RHO*SIN(THETA)
      CALL LOCATE(4,X,Y,LINEPZ)
      CALL NEWLIN
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,105)POLZRO(LINEPZ,3),POLZRO(LINEPZ,4)
105   FORMAT(9X,29HDELETED COMPLEX POLE WILL BE ,F10.7,6H +/-   ,F10.7)
      CALL RECOVR
      CALL ANGLE(727,260)
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,106)
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)
      CALL RECOVR
      CALL SCURSR(ICHAR,IX,IY)
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10
      CALL UPDATE(4,LINEPZ)
      CALL NUC
      RETURN
      END

      SUBROUTINE DRZPLR
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      DIMENSION LTPDRZ(64)
      DATA LTPDRZ/4HTO  D,4HELET,4HE A ,4HREAL,4H ZER,4HO FR,4HOM T
     X     ,4HHE  ,4HDISP,4HLAY,,4H ENT,4HER T,4HHE ',4HRHO',4H COO
     X     ,4HRDIN,4HATE ,4HASSO,4HCIAT,4HED W,4HITH ,4HA TH,4HETA
     X     ,4HOF  ,4HZERO,4H, OF,4H A P,4HOINT,4H NEA,4HR TH,4HE RO
     X     ,4HOT  ,4HTO B,4HE DE,4HLETE,4HD. E,4HNTER,4H THE,4H VAL
     X     ,4HUE  ,4HWITH,4HIN T,4HHE B,4HOX P,4HROVI,4HDED.,4H INC
     X     ,4HLUDE,4HA DE,4HCIMA,4HL, A,4HND M,4HINUS,4H SIG,4HN AS
     X     ,4H    ,4HAPPR,4HOPRI,4HATE.,4H NO ,4HSIGN,4H IS ,4HPOSI/
      DATA LTPDRZ(64)/4HTIVE/
      CALL CHCKSZ(-1,1)
      IF(IERROR.NE.0)RETURN
10    CALL ERASE
      CALL MOVABS(0,528)
      DO 30 I=1,57,8
      CALL ANMODE
      WRITE(6,100)(LTPDRZ(I+J-1),J=1,8)
      CALL RECOVR
30    CALL NEWLIN
100   FORMAT (21X,8A4)
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,101)
101   FORMAT(27X,'RHO')
```

```
      CALL RECOVR
      CALL NEWLIN
      CALL MOVABS(320,302)
      CALL DRWREL(0,22)
      CALL DRWREL(138,0)
      CALL DRWREL(0,-22)
      CALL DRWREL(-138,0)
      CALL MOVREL(0,3)
      CALL CARTN
      CALL ANMODE
      WRITE (6,103)
103   FORMAT (22X,'>')
      READ (5,104) RHO
104   FORMAT(F10.7,5X,F10.7)
      CALL RECOVR
      CALL LOCATE(1,RHO,0.,LINEPZ)
      CALL NEWLIN
      CALL NEWLIN
      CALL ANMODE
      THETA=0.
      WRITE(6,105)POLZRO(LINEPZ,3),POLZRO(LINEPZ,4)
105   FORMAT( 12X,26HDELETED REAL ZERO WILL BE ,F10.7,3X,F10.7)
      CALL RECOVR
      CALL ANGLE(677,260)
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,106)
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)
      CALL RECOVR
      CALL SCURSR(ICHAR,IX,IY)
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10
      CALL UPDATE(1,LINEPZ)
      CALL NUC
      RETURN
      END

      SUBROUTINE DRPPLR
      COMMON IXXXXX(4),POLZRO(20,5),IERROR
      DIMENSION LTPDRP(64)
      DATA LTPDRP/4HTO D,4HELET,4HE A ,4HREAL,4H POL,4HE FR,4HOM T
     X     ,4HHE  ,4HUNIT,4H CIR,4HCLE,,4H ENT,4HER T,4HHE ',4HRHO'
     X     ,4H    ,4HCOOR,4HDINA,4HTE F,4HOR A,4H 'TH,4HETA',4H VAL
     X     ,4HUE  ,4HOF Z,4HERO,,4H OF ,4HA PO,4HINT ,4HNEAR,4H THE
     X     ,4H    ,4HROOT,4H TO ,4HBE D,4HELET,4HED. ,4HENTE,4HR TH
     X     ,4HE   ,4HVALU,4HE WI,4HTHIN,4H THE,4H BOX,4H PRO,4HVIDE
     X     ,4HD.  ,4HINCL,4HUDE ,4HA DE,4HCIMA,4HL, A,4HND M,4HINUS
     X     ,4H    ,4HSIGN,4H AS ,4HAPPR,4HOPRI,4HATE.,4H    ,4H    /
      DATA LTPDRP(64)/4H    /
```

```
      CALL CHCKSZ(-1,3)                                                 ASDF4537
      IF(IERROR.NE.0)RETURN                                             ASDF4538
10    CALL ERASE                                                        ASDF4539
      CALL MOVABS(0,528)                                                ASDF4540
      DO 30 I=1,57,8                                                    ASDF4541
      CALL ANMODE                                                       ASDF4542
      WRITE(6,100)(LTPDRP(I+J-1),J=1,8)                                 ASDF4543
      CALL RECOVR                                                       ASDF4544
30    CALL NEWLIN                                                       ASDF4545
100   FORMAT (21X,8A4)                                                  ASDF4546
      CALL NEWLIN                                                       ASDF4547
      CALL ANMODE                                                       ASDF4548
      WRITE (6,101)                                                     ASDF4549
101   FORMAT(27X,'RHO')                                                 ASDF4550
      CALL RECOVR                                                       ASDF4551
      CALL NEWLIN                                                       ASDF4552
      CALL MOVABS(320,302)                                              ASDF4553
      CALL DRWREL(0,22)                                                 ASDF4554
      CALL DRWREL(138,0)                                                ASDF4555
      CALL DRWREL(0,-22)                                                ASDF4556
      CALL DRWREL(-138,0)                                               ASDF4557
      CALL MOVREL(0,3)                                                  ASDF4558
      CALL CARTN                                                        ASDF4559
      CALL ANMODE                                                       ASDF4560
      WRITE (6,103)                                                     ASDF4561
103   FORMAT (22X,'>')                                                  ASDF4562
      READ (5,104) RHO                                                  ASDF4563
104   FORMAT(F10.7)                                                     ASDF4564
      CALL RECOVR                                                       ASDF4565
      CALL NEWLIN                                                       ASDF4566
      CALL LOCATE(3,RHO,0.,LINEPZ)                                      ASDF4567
      CALL NEWLIN                                                       ASDF4568
      CALL ANMODE                                                       ASDF4569
      THETA=0.                                                          ASDF4570
      WRITE(6,105)POLZRO(LINEPZ,3),POLZRO(LINEPZ,4)                     ASDF4571
105   FORMAT(12X,26HDELETED REAL POLE WILL BE ,F10.7,3X,F10.7)          ASDF4572
      CALL RECOVR                                                       ASDF4573
      CALL ANGLE(677,260)                                               ASDF4574
      CALL NEWLIN                                                       ASDF4575
      CALL ANMODE                                                       ASDF4576
      WRITE(6,106)                                                      ASDF4577
106   FORMAT(21X,32HIF CORRECT TYPE 'Y', IF NOT 'N'.)                   ASDF4578
      CALL RECOVR                                                       ASDF4579
      CALL SCURSR(ICHAR,IX,IY)                                          ASDF4580
      IF(ICHAR.NE.121.AND.ICHAR.NE.89)GO TO 10                          ASDF4581
      CALL UPDATE(3,LINEPZ)                                             ASDF4582
      CALL NUC                                                          ASDF4583
      RETURN                                                            ASDF4584
```

```
      END                                                               ASDF4585
                                                                        ASDF4586
      SUBROUTINE ANGLE(IX,IY)                                           ASDF4587
      CALL MOVABS(IX,IY)                                                ASDF4588
      CALL MOVREL(14,20)                                                ASDF4589
      CALL DRWREL(-9,-18)                                               ASDF4590
      CALL DRWREL(20,0)                                                 ASDF4591
      CALL MOVREL(-3,-1)                                                ASDF4592
      CALL DRWREL(-1,7)                                                 ASDF4593
      CALL DRWREL(-3,5)                                                 ASDF4594
      CALL DRWREL(-3,3)                                                 ASDF4595
      CALL DRWREL(-4,2)                                                 ASDF4596
      CALL MOVREL(0,-20)                                                ASDF4597
      RETURN                                                            ASDF4598
      END                                                               ASDF4599
                                                                        ASDF4600
      SUBROUTINE ERROR                                                  ASDF4601
      COMMON IROOTS(4),POLZRO(20,5), IERROR                             ASDF4602
      CALL LST                                                          ASDF4603
      CALL MOVABS(0,154)                                                ASDF4604
      CALL ANMODE                                                       ASDF4605
      GO TO (2,2,2,2,4,4,4,4,6,8,8,8,6,8,6,8,6),IERROR                  ASDF4606
2     WRITE(6,100)                                                      ASDF4607
100   FORMAT(21X,'MAXIMUM SYSTEM ORDER EXCEEDED.')                      ASDF4608
      GO TO 10                                                          ASDF4609
4     WRITE(6,101)                                                      ASDF4610
101   FORMAT(19X,'ROOTS NOT AVAILABLE TO BE DELETED.')                  ASDF4611
      GO TO 10                                                          ASDF4612
6     WRITE(6,102)                                                      ASDF4613
102   FORMAT(11X,'POLES MAY NOT BE ENTERED OUTSIDE THE UNIT CIRCLE.')   ASDF4614
      GO TO 10                                                          ASDF4615
8     WRITE(6,103)                                                      ASDF4616
103   FORMAT(10X,'ZEROS MUST BE WITHIN TEN UNITS OF THE UNIT CIRCLE.')  ASDF4617
10    CALL RECOVR                                                       ASDF4618
      CALL NEWLIN                                                       ASDF4619
      CALL ANMODE                                                       ASDF4620
      WRITE(6,104)                                                      ASDF4621
104   FORMAT(25X,'COMMAND NOT PROCESSED.')                              ASDF4622
      CALL RECOVR                                                       ASDF4623
      CALL NEWLIN                                                       ASDF4624
      CALL ANMODE                                                       ASDF4625
      WRITE(6,105)                                                      ASDF4626
105   FORMAT(11X,'WHEN YOU ARE READY TO RETURN HIT',                    ASDF4627
     1' ANY KEYBOARD KEY')                                              ASDF4628
      CALL RECOVR                                                       ASDF4629
      CALL SCURSR(ICHAR, IX, IY)                                        ASDF4630
      IERROR=0                                                          ASDF4631
      CALL NUC                                                          ASDF4632
```

```
      RETURN                                                            ASDF4633
      END                                                               ASDF4634
                                                                        ASDF4635
      SUBROUTINE LST                                                    ASDF4636
      COMMON IROOTS(4),POLZRO(20,5),IERROR                              ASDF4637
      DIMENSION IRTTP(5)                                                ASDF4638
      DATA IZ/1HZ/,IP/1HP/,IZP/2HZP/                                    ASDF4639
      DATA IRTTP/2H  ,2HRZ,2HCZ,2HRP,2HCP/                              ASDF4640
      CALL ERASE                                                        ASDF4641
      CALL HOME                                                         ASDF4642
      CALL ANMODE                                                       ASDF4643
      WRITE(6,102) (IZ,I=1,70)                                          ASDF4644
  102 FORMAT (1X,70A1)                                                  ASDF4645
      CALL RECOVR                                                       ASDF4646
      CALL NEWLIN                                                       ASDF4647
      CALL ANMODE                                                       ASDF4648
      WRITE (6,108)                                                     ASDF4649
  108 FORMAT (1X,'Z',5X,'REAL',6X,'Z',3X,'IMAGINARY',3X,'Z'             ASDF4650
     1,6X,'RHO',6X,'Z',5X,'THETA',5X,'Z RT Z')                          ASDF4651
      CALL NEWLIN                                                       ASDF4652
      CALL RECOVR                                                       ASDF4653
      CALL ANMODE                                                       ASDF4654
      WRITE (6,102) (IZ,I= 1,70)                                        ASDF4655
      CALL RECOVR                                                       ASDF4656
      CALL NEWLIN                                                       ASDF4657
      DO 10 I=1,10                                                      ASDF4658
      CALL ANMODE                                                       ASDF4659
      ITYPEP=POLZRO(I,5)+1.                                             ASDF4660
      IF(POLZRO(I,5).NE.2..AND.POLZRO(I,5).NE.4.)                       ASDF4661
     1WRITE(6,104) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)                  ASDF4662
      IF(POLZRO(I,5).EQ.2..OR.POLZRO (I,5).EQ.4.)                       ASDF4663
     1WRITE(6,103) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)                  ASDF4664
  103 FORMAT(' Z   ',F10.7,'  Z +/-',F10.7,' Z  ',F10.7,                ASDF4665
     13X,'Z +/-',F10.7,' Z ',A2,' Z')                                   ASDF4666
  104 FORMAT(' Z   ',F10.7,'  Z    ',F10.7,' Z  ',F10.7,                ASDF4667
     1'   Z    ',F10.7,' Z ',A2,' Z')                                   ASDF4668
      CALL RECOVR                                                       ASDF4669
      CALL NEWLIN                                                       ASDF4670
   10 CONTINUE                                                          ASDF4671
      CALL ANMODE                                                       ASDF4672
      WRITE(6,112) (IZP,I=1,35)                                         ASDF4673
  112 FORMAT (1X,35A2)                                                  ASDF4674
      CALL RECOVR                                                       ASDF4675
      CALL NEWLIN                                                       ASDF4676
      DO 20 I=11,20                                                     ASDF4677
      CALL ANMODE                                                       ASDF4678
      ITYPEP=POLZRO(I,5)+1.                                             ASDF4679
      IF (POLZRO(I,5).NE.2..AND.POLZRO(I,5).NE.4.)                      ASDF4680
```

```
     1WRITE(6,106) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)
      IF(POLZRO (I,5).EQ.2..OR.POLZRO(I,5).EQ.4.)
     1WRITE(6,105) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)
105   FORMAT(' P    ',F10.7,'  P +/-',F10.7,' P   ',F10.7,
     13X,'P +/-',F10.7,' P ',A2,' P')
106   FORMAT(' P    ',F10.7,'  P     ',F10.7,' P   ',F10.7,
     1'    P     ',F10.7,' P ',A2,' P')
      CALL RECOVR
      CALL NEWLIN
20    CONTINUE
      CALL ANMODE
      WRITE(6,102) (IP, I =1,70)
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,107)
107   FORMAT(//)
      CALL RECOVR
      IF(IERROR.NE.0)RETURN
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,110) IROOTS(1)
110   FORMAT(20X,'NUMBER OF REAL ZEROS:',I3)
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,111) IROOTS(2)
111   FORMAT(20X,'NUMBER OF COMPLEX ZERO PAIRS:',I3)
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,113) IROOTS(3)
113   FORMAT (20X,'NUMBER OF REAL POLES:',I3)
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      WRITE (6,114) IROOTS(4)
114   FORMAT (20X, 'NUMBER OF COMPLEX POLE PAIRS:',I3)
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      WRITE(6,115)
115   FORMAT(' HIT SPACE THEN RETURN')
      CALL RECOVR
      CALL NEWLIN
      CALL ANMODE
      READ(5,116) IN
116   FORMAT(A1)
```

```
      CALL RECOVR
      CALL NUC
      RETURN
      END

      SUBROUTINE PLTTBL
C*
C* SUBROUTINE PLOTS THE POLZRO TABLE OF THE UNIT CIRCLE  (12)
C*
      COMMON IXXXXX(4),POLZRO(20,5)
      DO 10 I=1,20
C*
C* IF POLZRO(I,5)=0 NO ROOT HAS BEEN STORED
C*
      IF(POLZRO(I,5).EQ.0.)GO TO 10
      CALL PLOTRT(IFIX(POLZRO(I,5)),POLZRO(I,1),POLZRO(I,2))
   10 CONTINUE
      RETURN
      END

      SUBROUTINE CHCKSZ (IDELTA, ITYPE)
C*
C* THIS SUBROUTINE CHECKS THE TABLE OF ROOTS TO SEE
C* IF IT IS POSSIBLE TO EXECUTE THE DESIRED SUBROUTINE
C*
C* IDELTA: IS THE CHANGE IN THE NUMBER OF ROOTS EXPECTED
C*        BY THE CALLING SUBROUTINE
C* ITYPE: IS THE TYPE OF ROOTS TO BE ALTERED, I.E.,
C*        COMPLEX, REAL, ZERO, OR POLE
C* IERROR: RETURNS THE ERROR MESSAGE NUMBER IF IT IS
C*        NOT POSSIBLE TO EXECUTE THE SELECTED COMMAND;
C*        THE MAXIMUM SYSTEM ORDER IS LIMITED TO TEN AND
C*        ROOTS MAY NOT BE DELETED IF THEY HAVE NOT BEEN INPUT
C*
      COMMON IROOTS(4),PXXXXX(20,5),IERROR
C*
C* IROOTS(1) = FLAG COUNTING REAL ZEROS
C* IROOTS(2) = FLAG COUNTING COMPLEX ZERO PAIRS
C* IROOTS(3) = FLAG COUNTING REAL POLES
C* IROOTS(4) = FLAG COUNTING COMPLEX POLE PAIRS
C*
C* DETERMINE WHETHER ROOT(S) ARE BEING ADDED OR DELETED (1)
C*
      IF(IDELTA.LT.0)GO TO 10
C*
C* ROOT(S) ARE BEING ADDED  (3)
C* CALCULATE THE EXPECTED ORDER OF THE SYSTEM  (1)
C*
```

```
      IF(ITYPE.EQ.1.OR.ITYPE.EQ.2)NORDER=IROOTS(1)+2*IROOTS(2)+IDELTA
      IF(ITYPE.EQ.3.OR.ITYPE.EQ.4)NORDER=IROOTS(3)+2*IROOTS(4)+IDELTA
C*
C* IF "NORDER" IS GREATER THAN 10, MAXIMUM SYSTEM ORDER HAS BEEN
C* EXCEEDED AND AN ERROR RESULTS  (1)
C*
      IF(NORDER.GT.10)IERROR=ITYPE
 20   IF(Y.GT.0.)CALL POINTA(XX,YY)
      RETURN
C*
C* ROOTS ARE BEING DELETED  (2)
C* CALCULATE THE EXPECTED ROOT TOTAL IF THE CALLING
C* SUBROUTINE EXECUTES  (1)
C* PLUS SIGN IS DUE TO "IDELTA" BEING LESS THAN ZERO
C*
 10   IXPTOT=IROOTS(ITYPE)+IDELTA
C*
C* IF THE EXPECTED ROOTS TOTAL "IXPTOT" IS NEGATIVE
C* AN ERROR IS GENERATED  (1)
C*
      IF(IXPTOT.LT.0)IERROR=ITYPE+4
 999  RETURN
      END

      SUBROUTINE UPDATE(ITYPE,LINEPZ)
C*
C* THIS SUBROUTINE WILL UPDATE THE POLZRO TABLE WHEN A ROOT IS REMOVED
C*
      COMMON IROOTS(4),POLZRO(20,5)
C*
C* "LINEPZ" IS THE LINE OF POLZRO ARRAY TO BE DELETED
C*
      IF(LINEPZ.EQ.20)GO TO 45
      IF(LINEPZ.GT.10)GO TO 30
      IF(LINEPZ.EQ.10)GO TO 15
      DO 10 J=LINEPZ,9
      DO 10 K=1,5
 10   POLZRO(J,K)=POLZRO(J+1,K)
 15   DO 20 M=1,4
 20   POLZRO(10,M)=-10.
      POLZRO(10,5)=0.
      GO TO 60
 30   DO 40 J=LINEPZ,19
      DO 40 K=1,5
 40   POLZRO(J,K)=POLZRO(J+1,K)
 45   DO 50 M=1,4
 50   POLZRO(20,M)=-10.
      POLZRO(20,5)=0.
```

```
C*
C* ADJUST THE ROOT COUNTER SINCE A ROOT HAS BEEN DELETED
C*
 60    IROOTS(ITYPE)=IROOTS(ITYPE)-1
       RETURN
       END

       SUBROUTINE LOCATE(ITYPE,X,Y,LINEPZ)
C*
C* THIS SUBROUTINE LOCATES THE STORED ROOT NEAREST THE CURSOR POSITION
C*
       COMMON IXXXXX(4), POLZRO(20,5)
C*
C* "ITYPE" IS THE TYPE OF ROOT SEARCHED
C* "LINEPZ" IS THE LINE OF POLZRO WHICH CONTAINS THE ROOT
C*     OF THE DESIRED TYPE WHICH IS CLOSEST TO THE
C*     USER-POSITIONED CURSOR
C* "DIST" IS THE SHORTEST DISTANCE FOUND FROM STORED ROOT
C*     TO THE USER-POSITIONED CURSOR
C*
C* STRRTS STORES ONLY THE COMPLEX POLE IN THE TOP HALF PLANE
C*
       SY=ABS(Y)
       DIST=20.
       DO 10 I=1,20
       IF(IFIX(POLZRO(I,5)).NE.ITYPE)GO TO 10
       CKDIST=SQRT((X-POLZRO(I,1))**2+(SY-POLZRO(I,2))**2)
       IF(CKDIST.LT.DIST)LINEPZ=I
       IF(CKDIST.LT.DIST)DIST=CKDIST
 10    CONTINUE
       RETURN
       END

       SUBROUTINE START
C*
C* THIS SUBROUTINE INITIALIZES ALL THE VARIABLES STORED IN
C* COMMON BEFORE EXECUTION OF THE PROGRAM  (10)
C*
       COMMON IROOTS(4),POLZRO(20,5),IERROR,IOLD
       IOLD=0
       DO 10 I=1,4
 10    IROOTS(I)=0
       IERROR=0
       DO 20 J=1,20
       POLZRO(J,5)=0.
       DO 20 K=1,4
 20    POLZRO(J,K)=-10.
       RETURN
```

```
      END                                                               ASDF4873
                                                                        ASDF4874
      SUBROUTINE STRRTS(ITYPE,MODE,ARGMT1,ARGMT2)                       ASDF4875
C*                                                                      ASDF4876
C* THIS SUBROUTINE IS DESIGNED TO STORE ROOTS PREVIOUSLY                ASDF4877
C* GENERATED IN THE ROOT TABLE "POLZRO"                                 ASDF4878
C*                                                                      ASDF4879
      COMMON IROOTS(4), POLZRO(20,5), IERROR                            ASDF4880
      PI=3.14159265                                                     ASDF4881
      IF(ARGMT1.EQ.0..AND.ARGMT2.EQ.0.)GO TO 45                         ASDF4882
C*                                                                      ASDF4883
C* GIVEN "MODE", "ARGMT1", AND "ARGMT2", GENERATE THE REAL,             ASDF4884
C* IMAGINARY, RADIUS, AND THETA VALUES OF THE UPPER HALF PLANE          ASDF4885
C* ROOT TO BE STORED (THIS INCLUDES THE REAL AXIS).                     ASDF4886
C*                                                                      ASDF4887
      GO TO (10,20),MODE                                                ASDF4888
C*                                                                      ASDF4889
C* "MODE" DICTATES THAT "ARGMT1" AND "ARGMT2" ARE RECTANGULAR           ASDF4890
C*                                                                      ASDF4891
 10   RREAL=ARGMT1                                                      ASDF4892
      RIMAG=ABS(ARGMT2)                                                 ASDF4893
      RADIUS=SQRT(ARGMT1**2+ARGMT2**2)                                  ASDF4894
      THETA=ABS(ATAN2(ARGMT1,ARGMT2))                                   ASDF4895
      GO TO 30                                                          ASDF4896
C*                                                                      ASDF4897
C* "MODE" DICTATES THAT "ARGMT1" AND "ARGMT2" ARE POLAR                 ASDF4898
C*                                                                      ASDF4899
 20   RADIUS=ARGMT1                                                     ASDF4900
 50   IF(ARGMT2.LE.PI.AND.ARGMT2.GT.-PI)GO TO 40                        ASDF4901
      IF(ARGMT2.LE.-PI)ARGMT2=ARGMT2+2.*PI                              ASDF4902
      IF(ARGMT2.GT.PI)ARGMT2=ARGMT2-2.*PI                               ASDF4903
      GO TO 50                                                          ASDF4904
 40   THETA=ARGMT2                                                      ASDF4905
      RREAL=RADIUS*COS(THETA)                                           ASDF4906
      RIMAG=RADIUS*SIN(THETA)                                           ASDF4907
      GO TO 30                                                          ASDF4908
C*                                                                      ASDF4909
C* IF THE ROOT IS TO BE AT THE ORIGIN,                                  ASDF4910
C* ASSIGN VALUES SUCH THAT ILLEGAL LIBRARY FUNCTION                     ASDF4911
C* ARGUMENTS WILL NOT BE ASSIGNED                                       ASDF4912
C*                                                                      ASDF4913
 45   RREAL=0.                                                          ASDF4914
      RIMAG=0.                                                          ASDF4915
      RADIUS=0.                                                         ASDF4916
      THETA=0.                                                          ASDF4917
C*                                                                      ASDF4918
C* STORAGE PROCEDURE IS A FUNCTION OF THE TYPE OF ROOTS TO BE STORED    ASDF4919
C*                                                                      ASDF4920
```

```
C*              RZ  CZ  RP  CP                                          ASDF4921
 30    GO TO (110,210,120,220), ITYPE                                   ASDF4922
C*                                                                      ASDF4923
C* A REAL ZERO IS BEING STORED                                          ASDF4924
C* IROOTS(1) IS THE NUMBER OF REAL ZEROS ALREADY STORED                 ASDF4925
C*                                                                      ASDF4926
 110   POLZRO(IROOTS(1)+IROOTS(2)+1,1)=RREAL                            ASDF4927
       POLZRO(IROOTS(1)+IROOTS(2)+1,2)=0.                               ASDF4928
       POLZRO(IROOTS(1)+IROOTS(2)+1,3)=ABS(RREAL)                       ASDF4929
       IF(ARGMT1.GE.0.)THETA=0.                                         ASDF4930
       IF(ARGMT1.LT.0.)THETA=PI                                         ASDF4931
       POLZRO(IROOTS(1)+IROOTS(2)+1,4)=THETA                            ASDF4932
       POLZRO(IROOTS(1)+IROOTS(2)+1,5)=1.                               ASDF4933
       IROOTS(1)=IROOTS(1)+1                                            ASDF4934
       GO TO 999                                                        ASDF4935
C*                                                                      ASDF4936
C* A REAL POLE IS TO BE STORED                                          ASDF4937
C* IROOTS(3) IS THE NUMBER OF REAL POLES ALREADY STORED                 ASDF4938
C*                                                                      ASDF4939
 120   POLZRO(IROOTS(3)+IROOTS(4)+11,1)=RREAL                           ASDF4940
       POLZRO(IROOTS(3)+IROOTS(4)+11,2)=0.                              ASDF4941
       POLZRO(IROOTS(3)+IROOTS(4)+11,3)=ABS(RREAL)                      ASDF4942
       IF(ARGMT1.GE.0.)THETA=0.                                         ASDF4943
       IF(ARGMT1.LT.0.)THETA=PI                                         ASDF4944
       POLZRO(IROOTS(3)+IROOTS(4)+11,4)=THETA                           ASDF4945
       POLZRO(IROOTS(3)+IROOTS(4)+11,5)=3.                              ASDF4946
       IROOTS(3)=IROOTS(3)+1                                            ASDF4947
       GO TO 999                                                        ASDF4948
C*                                                                      ASDF4949
C* A COMPLEX ZERO IS TO BE STORED                                       ASDF4950
C* IROOTS(2) IS THE NUMBER OF COMPLEX ZERO PAIRS ALREADY STORED         ASDF4951
C*                                                                      ASDF4952
 210   POLZRO(IROOTS(2)+IROOTS(1)+1,1)=RREAL                            ASDF4953
       POLZRO(IROOTS(2)+IROOTS(1)+1,2)=ABS(RIMAG)                       ASDF4954
       POLZRO(IROOTS(2)+IROOTS(1)+1,3)=ABS(RADIUS)                      ASDF4955
       IF(ARGMT1.EQ.0..AND.ARGMT2.EQ.0.)RADIUS=1.                       ASDF4956
       POLZRO(IROOTS(2)+IROOTS(1)+1,4)=ARCOS(RREAL/ABS(RADIUS))         ASDF4957
       POLZRO(IROOTS(2)+IROOTS(1)+1,5)=2.                               ASDF4958
       IROOTS(2)=IROOTS(2)+1                                            ASDF4959
       GO TO 999                                                        ASDF4960
C*                                                                      ASDF4961
C* A COMPLEX POLE IS TO BE STORED                                       ASDF4962
C* IROOTS(4) IS THE NUMBER OF COMPLEX POLE PAIRS ALREADY STORED         ASDF4963
C*                                                                      ASDF4964
 220   POLZRO(IROOTS(4)+IROOTS(3)+11,1)=RREAL                           ASDF4965
       POLZRO(IROOTS(4)+IROOTS(3)+11,2)=RIMAG                           ASDF4966
       POLZRO(IROOTS(4)+IROOTS(3)+11,3)=ABS(RADIUS)                     ASDF4967
       IF(ARGMT1.EQ.0..AND.ARGMT2.EQ.0.)RADIUS=1.                       ASDF4968
```

```
      POLZRO(IROOTS(4)+IROOTS(3)+11,4)=ARCOS(RREAL/ABS(RADIUS))
      POLZRO(IROOTS(4)+IROOTS(3)+11,5)=4.
      IROOTS(4)=IROOTS(4)+1
  999 IF(ITYPE.EQ.1.OR.ITYPE.EQ.3)CALL PLOTRT(ITYPE,RREAL,0.)
      IF(ITYPE.EQ.2.OR.ITYPE.EQ.4)CALL PLOTRT(ITYPE,RREAL,RIMAG)
      RETURN
      END

      SUBROUTINE PLOTRT (ITYPE, RREAL, RIMAG)
C*
C* THE PURPOSE OF THIS SUBROUTINE IS TO PLOT
C* THE ROOT JUST ENTERED INTO THE SYSTEM
C*
      DIMENSION IZERO(16),ISTAR(16)
C*
C* DATA POINTS REQUIRED FOR DRAWING ZEROS
C*
      DATA IZERO/2,-2,0,-2,-2,-2,-2,0,-2,2,0,2,2,2,2,0/
      DATA ISTAR/-5,5,10,-10,-5,10,0,-10,5,10,-10,-10,10,5,-10,0/
C*
C*  IF ZERO IS OUTSIDE THE UNIT CIRCLE, PLOT A STAR.
C*
      RADIUS = SQRT(RREAL**2 + RIMAG**2)
      IF(RADIUS.GT.1.)GO TO 300
C*  SINCE ROOT IS INSIDE THE UNIT CIRCLE
C*
C* IF THE ROOT IS A ZERO GO TO 100
C* IF THE ROOT IS A POLE GO TO 200
C*
C*             RZ  CZ  RP  CP
      GO TO (100,100,200,200),ITYPE
C*
C* CENTER THE ZERO ON THE LOCATION STORED IN POLZRO
C*
  100 CALL MOVEA(RREAL,RIMAG)
C*
C* GENERATE THE ZERO  (3)
C*
      CALL MOVREL(1,3)
      DO 120 I=1,15,2
  120 CALL DRWREL(IZERO(I),IZERO(I+1))
C*
C* IF THE ZERO IS REAL, RETURN
C*
      IF(ITYPE.EQ.1)GO TO 999
C*
C* PLOT THE CORRESPONDING COMPLEX ZERO IN THE LOWER HALF PLANE  (4)
C*
```

```
      CALL MOVEA(RREAL,-RIMAG)                                          ASDF5017
      CALL MOVREL(1,3)                                                  ASDF5018
      DO 130 I=1,15,2                                                   ASDF5019
  130 CALL DRWREL (IZERO(I),IZERO(I+1))                                 ASDF5020
      GO TO 999                                                         ASDF5021
C*                                                                      ASDF5022
C* PLOT THE POLE (REAL, OR UPPER HALF PLANE)  (5)                       ASDF5023
C*                                                                      ASDF5024
  200 CALL MOVEA (RREAL,RIMAG)                                          ASDF5025
      CALL MOVREL(3,3)                                                  ASDF5026
      CALL DRWREL(-6,-6)                                                ASDF5027
      CALL MOVREL(0,6)                                                  ASDF5028
      CALL DRWREL (6,-6)                                                ASDF5029
C*                                                                      ASDF5030
C* IF THE POLE IS REAL, RETURN                                          ASDF5031
C*                                                                      ASDF5032
      IF(ITYPE.EQ.3)GO TO 999                                           ASDF5033
C*                                                                      ASDF5034
C* IF THE POLE IS COMPLEX, PLOT THE CORRESPONDING                       ASDF5035
C* LOWER HALF PLANE POLE  (5)                                           ASDF5036
C*                                                                      ASDF5037
      CALL MOVEA (RREAL,-RIMAG)                                         ASDF5038
      CALL MOVREL(3,3)                                                  ASDF5039
      CALL DRWREL(-6,-6)                                                ASDF5040
      CALL MOVREL(0,6)                                                  ASDF5041
      CALL DRWREL(6,-6)                                                 ASDF5042
      GO TO 999                                                         ASDF5043
  300 ANGLE=ATAN2(RIMAG,RREAL)                                          ASDF5044
      PREAL=1.05*COS(ANGLE)                                             ASDF5045
      PIMAG=1.05*SIN(ANGLE)                                             ASDF5046
      CALL MOVEA(PREAL,PIMAG)                                           ASDF5047
      DO 320 M=1,2                                                      ASDF5048
      DO 310 I=1,16,4                                                   ASDF5049
      CALL MOVREL(ISTAR(I),ISTAR(I+1))                                  ASDF5050
      CALL DRWREL(ISTAR(I+2),ISTAR(I+3))                                ASDF5051
  310 CONTINUE                                                          ASDF5052
      IF(M.EQ.2.OR.ITYPE.EQ.1)GO TO 999                                 ASDF5053
  320 CALL MOVEA(PREAL,-PIMAG)                                          ASDF5054
  999 RETURN                                                            ASDF5055
      END                                                               ASDF5056
```

APPENDIX H: SOURCE DECK FOR ASDF COMMAND: RESPONSE

```
C*****************************************************          ASDF6001
C*                                                    *          ASDF6002
C*      SOURCE DECK FOR ASDF COMMAND:  RESPONSE       *          ASDF6003
C*                                                    *          ASDF6004
C*****************************************************          ASDF6005
      COMMON IROOTS(4),POLZRO(20,5),IERROR                       ASDF6006
      COMMON /ABVCTR/ A(11), B(11)                               ASDF6007
      COMMON /IMPULS/ YI(1025), YS(1025), EN(1025)               ASDF6008
      COMMON /FREQ/ YP(1026), YM(1026), PI(1026)                 ASDF6009
      LINEAR=1                                                   ASDF6010
      LOG=2                                                      ASDF6011
C*                                                               ASDF6012
C* INITIALIZE A AND B TO ZERO                                    ASDF6013
C*                                                               ASDF6014
      DO 5 K=1,20                                                ASDF6015
      A(K)=0.                                                    ASDF6016
 5    B(K)=0.                                                    ASDF6017
C*                                                               ASDF6018
C*                                                               ASDF6019
C* READ THE POLZRO TABLE OFF OF THE FILE                         ASDF6020
C*                                                               ASDF6021
      DO 20 I=1,20                                               ASDF6022
100   FORMAT(1X,5F12.8)                                          ASDF6023
20    READ(1,100) (POLZRO(I,J),J=1,5)                            ASDF6024
C*                                                               ASDF6025
C* READ THE IROOTS VALUES FROM THE FILE                          ASDF6026
C*                                                               ASDF6027
      READ(1,101) (IROOTS(I),I=1,4)                              ASDF6028
101   FORMAT(1X,4I2)                                             ASDF6029
      READ(1,102) FLTRGN                                         ASDF6030
102   FORMAT(1X,F12.8)                                           ASDF6031
      FLTRGN=ABS(FLTRGN)                                         ASDF6032
      ICHAR=0                                                    ASDF6033
C*                                                               ASDF6034
C* PRINT OUT A TABLE OF THE POLES AND ZEROS                      ASDF6035
C*                                                               ASDF6036
      CALL FLTR2(FLTRGN)                                         ASDF6037
C*                                                               ASDF6038
C* GENERATE THE A(K) COEFFICIENTS                                ASDF6039
C*                                                               ASDF6040
      CALL ASUBK(A,NORDER)                                       ASDF6041
      NDEN=NORDER+1                                              ASDF6042
C*                                                               ASDF6043
C* GENERATE THE B(R) COEFFICIENTS                                ASDF6044
C*                                                               ASDF6045
      CALL BSUBR(B,NORDER)                                       ASDF6046
      NNUM=NORDER+1                                              ASDF6047
C*                                                               ASDF6048
```

```
C* CHECK TO SEE IF THE FILTER IS CAUSAL
C*
      IF(NNUM.GT.NDEN)WRITE(6,103)
 103  FORMAT(10(/),' FILTER IS NOT CAUSAL - PROGRAM TERMINATES')
      IF(NNUM.GT.NDEN)STOP
C*
C* ADJUST THE COEFFICIENTS IF NNUM DOES NOT EQUAL NDEN
C*
      IF(NNUM.LT.NDEN)CALL ADJUST(NNUM,NDEN)
C*
C* COMPUTE THE UNIT SAMPLE RESPONSE
C*
      CALL RSPNSE(0,FLTRGN)
C*
C* PLOT THE UNIT SAMPLE RESPONSE
C*
      CALL PLTIMP(ICHAR)
      IF(ICHAR.EQ.88.OR.ICHAR.EQ.120)GO TO 30
C*
C* CALCULATE THE STEP RESPONSE
C*
      CALL RSPNSE(1,FLTRGN)
C*
C* PLOT THE STEP RESPONSE
C*
      CALL PLTSTP(ICHAR)
      IF(ICHAR.EQ.88.OR.ICHAR.EQ.120)GO TO 30
C*
C* COMPUTE THE MAGNITUDE AND PHASE OF THE TRANSFER FUNCTION
C*
      CALL FRQNCY(NNUM,NDEN,FLTRGN)
C*
C* PLOT THE PHASE OF THE TRANSFER FUNCTION
C*
      CALL PLTTRF(3,LINEAR,ICHAR)
      IF(ICHAR.EQ.88.OR.ICHAR.EQ.120)GO TO 30
C*
C* PLOT THE MAGNITUDE OF THE TRANSFER FUNCTION
C* FIRST WITH LINEAR AXIS, THEN IN DECIBELS
C*
      CALL PLTTRF(4,LINEAR,ICHAR)
      IF(ICHAR.EQ.88.OR.ICHAR.EQ.120)GO TO 30
      CALL PLTTRF(4,LOG,ICHAR)
      IF(ICHAR.EQ.88.OR.ICHAR.EQ.120)GO TO 30
 30   STOP
      END

      SUBROUTINE ASUBK (POLE,NORDER)
```

```
C*                                                                      ASDF6097
C* GENERATE THE AK COEFFICIENTS FROM THE POLES                          ASDF6098
C*                                                                      ASDF6099
      COMMON IROOTS(4), POLZRO(20,5), IERROR                            ASDF6100
      COMPLEX POLES(11), ACOEF(11)                                      ASDF6101
      REAL POLE(11)                                                     ASDF6102
C*                                                                      ASDF6103
C* INITIALIZE ALL THE VECTORS AND INDICES TO ZERO                        ASDF6104
C*                                                                      ASDF6105
      DO 5 K=1,11                                                       ASDF6106
      POLES(K)=(0.,0.)                                                  ASDF6107
      POLE(K)=0.                                                        ASDF6108
 5    ACOEF(K)=0.                                                       ASDF6109
C*                                                                      ASDF6110
C* "J" INDEXES THE TABLE "POLES"                                        ASDF6111
C* "K" INDEXES THE TABLE "POLZRO"                                       ASDF6112
C*                                                                      ASDF6113
      J=1                                                               ASDF6114
      K=1                                                               ASDF6115
C*                                                                      ASDF6116
C* COMPUTE THE ORDER OF THE SYSTEM OF POLES                             ASDF6117
C*                                                                      ASDF6118
      NORDER=IROOTS(3)+2*IROOTS(4)                                      ASDF6119
      IF(NORDER.NE.0)GO TO 10                                           ASDF6120
      POLE(1)=1.0                                                       ASDF6121
      RETURN                                                            ASDF6122
C*                                                                      ASDF6123
C* THE VECTOR "POLES" WILL CONTAIN A LIST OF ALL                        ASDF6124
C* THE POLES, IE., X+J0, X+JY, X-JY, . . .                              ASDF6125
C*                                                                      ASDF6126
 10   POLES(J)=CMPLX(POLZRO(10+K,1),POLZRO(10+K,2))                     ASDF6127
C*                                                                      ASDF6128
C* TEST TO SEE IF ALL ROOTS HAVE BEEN                                   ASDF6129
C* STORED IN THE "POLES" VECTOR                                         ASDF6130
C*                                                                      ASDF6131
      IF(J.EQ.NORDER)GO TO 20                                           ASDF6132
      J=J+1                                                             ASDF6133
C*                                                                      ASDF6134
C* IF THE ROOT IS REAL CONTINUE PROCESING THE NEXT ROOT                 ASDF6135
C*                                                                      ASDF6136
      IF(POLZRO(K+10,5).NE.3)GO TO 15                                   ASDF6137
      K=K+1                                                             ASDF6138
      GO TO 10                                                          ASDF6139
C*                                                                      ASDF6140
C* SINCE THE ROOT IS COMPLEX PUT THE CONJUGATE                          ASDF6141
C* OF THE ROOT IN THE "POLES" TABLE.                                    ASDF6142
C*                                                                      ASDF6143
 15   POLES(J)=CMPLX(POLZRO(10+K,1),-POLZRO(10+K,2))                    ASDF6144
```

```
      IF(J.EQ.NORDER) GO TO 20                                          ASDF6145
      J=J+1                                                             ASDF6146
      K=K+1                                                             ASDF6147
      GO TO 10                                                          ASDF6148
C*                                                                      ASDF6149
C* GENERATE THE COEFFICIENTS FOR THE D(Z) POLYNOMIAL                    ASDF6150
C* OF THE TRANSFER FUNCTION H(Z).                                       ASDF6151
C*                                                                      ASDF6152
 20   CALL COEFF(POLES,NORDER,ACOEF)                                    ASDF6153
C*                                                                      ASDF6154
C* CHANGE THE COEFFICIENTS FROM COMPLEX ASUBK TO REAL ASUBK             ASDF6155
C*                                                                      ASDF6156
      DO 40 J=1,11                                                      ASDF6157
 40   POLE(J)=REAL(ACOEF(J))                                            ASDF6158
      RETURN                                                            ASDF6159
      END                                                               ASDF6160
                                                                        ASDF6161
      SUBROUTINE BSJBR(ZERO,NORDER)                                     ASDF6162
C*                                                                      ASDF6163
C* GENERATE THE B(R) COEFFICIENTS FROM THE ZEROS                        ASDF6164
C*                                                                      ASDF6165
      COMMON IROOTS(4),POLZRO(20,5),IERROR                              ASDF6166
      COMPLEX ZEROS(11), BCOEF(11)                                      ASDF6167
      REAL ZERO(11)                                                     ASDF6168
C*                                                                      ASDF6169
C* INITIALIZE ALL VECTORS AND INDICES                                   ASDF6170
C*                                                                      ASDF6171
      DO 5 K=1,11                                                       ASDF6172
      ZEROS(K)=(0.,0.)                                                  ASDF6173
      ZERO(K)=0.                                                        ASDF6174
 5    BCOEF(K)=(0.,0.)                                                  ASDF6175
C*                                                                      ASDF6176
C* "J" INDEXES THE "ZEROS" TABLE                                        ASDF6177
C* "K" INDEXES THE "POLZRO" TABLE                                       ASDF6178
C*                                                                      ASDF6179
      J=1                                                               ASDF6180
      K=1                                                               ASDF6181
C*                                                                      ASDF6182
C* COMPUTE THE ORDER OF THE SYSTEM OF ZEROS.                            ASDF6183
C*                                                                      ASDF6184
      NORDER=IROOTS(1)+2*IROOTS(2)                                      ASDF6185
      IF(NORDER.NE.0)GO TO 10                                           ASDF6186
      ZERO(1)=1.0                                                       ASDF6187
      RETURN                                                            ASDF6188
C*                                                                      ASDF6189
C* THE VECTOR "ZEROS" CONTAINS A LIST OF ALL                            ASDF6190
C* OF THE ZEROS, IE., X+J0, X+JY, X-JY, . . .                           ASDF6191
C*                                                                      ASDF6192
```

```
 10    ZEROS(J)=CMPLX(POLZRO(K,1),POLZRO(K,2))                          ASDF6193
C*                                                                      ASDF6194
C* CHECK TO SEE IF ALL THE ROOTS HAVE BEEN PROCESSED                    ASDF6195
C*                                                                      ASDF6196
       IF(J.EQ.NORDER)GO TO 20                                          ASDF6197
       J=J+1                                                            ASDF6198
C*                                                                      ASDF6199
C* IF THE ROOT IS REAL CONTINUE PROCESSING THE NEXT ROOT                 ASDF6200
C*                                                                      ASDF6201
       IF(POLZRO(K,5).NE.1)GO TO 15                                     ASDF6202
       K=K+1                                                            ASDF6203
       GO TO 10                                                         ASDF6204
C*                                                                      ASDF6205
C* SINCE THE ROOT IS COMPLEX, ENTER THE CONJUGATE                        ASDF6206
C* INTO THE TABLE "ZEROS".                                              ASDF6207
C*                                                                      ASDF6208
 15    ZEROS(J)=CMPLX(POLZRO(K,1),-POLZRO(K,2))                         ASDF6209
       IF(J.EQ.NORDER)GO TO 20                                          ASDF6210
       J=J+1                                                            ASDF6211
       K=K+1                                                            ASDF6212
       GO TO 10                                                         ASDF6213
C*                                                                      ASDF6214
C* GENERATE THE BSUBR COEFFICIENTS                                      ASDF6215
C*                                                                      ASDF6216
 20    CALL COEFF(ZEROS, NORDER, BCOEF)                                 ASDF6217
C*                                                                      ASDF6218
C* MAKE THE COEFFICIENTS BSUBR REAL.                                    ASDF6219
C*                                                                      ASDF6220
       DO 40 J=1,11                                                     ASDF6221
 40    ZERO(J)=REAL(BCOEF(J))                                           ASDF6222
       RETURN                                                           ASDF6223
       END                                                              ASDF6224
                                                                        ASDF6225
       SUBROUTINE COEFF(V,N,H1)                                         ASDF6226
C*                                                                      ASDF6227
C* "V" - VECTOR OF ROOTS; "N" - THE ORDER OF THE SYSTEM                 ASDF6228
C* "H1" - VECTOR WITH THE FINAL COEFFICIENTS                            ASDF6229
C* H (1)=Z**(0), H1(2)=Z**(-1), . . . ,H1(11)=Z**(-10)                  ASDF6230
C*                                                                      ASDF6231
       COMPLEX H1(11),H2(11),H3(11),V(11)                               ASDF6232
C* INITIALIZE VECTORS                                                   ASDF6233
       IZERO=0                                                          ASDF6234
       DO 10 I=1,11                                                     ASDF6235
       H1(I)=(0.,0.)                                                    ASDF6236
       H2(I)=(0.,0.)                                                    ASDF6237
 10    H3(I)=(0.,0.)                                                    ASDF6238
       H1(1)=(1.,0.)                                                    ASDF6239
C*                                                                      ASDF6240
```

```
C* THE ROOTS COME FROM THE FORMER SUBROUTINE
C* THESE ARE ROOTS OF THE POLYNOMIALS N(Z), AND D(Z)
C* WHERE H(Z)=N(Z)/D(Z).
C* THE FACTORS OF THE POLYNOMIAL ARE OF THE FORM Z-ROOT
C*
      DO 15 K=1,11
 15   V(K)=-V(K)
C*
C* WHEN K=N, WE HAVE GENERATED THE H1 COEFFICIENTS
C* OF N(Z), OR D(Z) OF H(Z)=N(Z)/D(Z)
C* THE REMAINING ITERATIONS ARE PERFORMED TO SIMPLIFY
C* OBTAINING COEFFICIENTS FOR N(Z**-1), AND D(Z**-1) LATER
C*
      DO 30 K=1,10
C*
C* "H2" REPRESENTS THE PREVIOUS RESULT TIMES Z**(-1)
C*
      DO 40 I=1,10
 40   H2(I+1)=H1(I)
      H2(1)=(0.,0.)
      DO 50 I=1,11
 50   H3(I)=H1(I)*V(1)
C*
C* GENERATE THE NEW FINAL PRODUCT WITH K ROOTS MULTIPLIED.
C*
      DO 60 I=1,11
 60   H1(I)=H2(I)+H3(I)
C*
C* GET THE NEXT ROOT TO BE MULTIPLIED AND PUT IT INTO "V(1)"
C*
 65   DO 70 I=1,10
 70   V(I)=V(I+1)
 30   CONTINUE
C*
C* TO GET THE COEFFICIENTS FOR H(Z**-1)=N(Z**-1)/D(Z**-1)
C* THE ORDER OF THE COEFFICIENTS MUST BE REVERSED.
C*
      DO 80 I=1,11
 80   H2(I)=H1(12-I)
      DO 90 I=1,11
 90   H1(I)=H2(I)
 95   RETURN
      END

      SUBROUTINE ADJUST(NNUM,NDEN)
      COMMON /ABVCTR/A(11),B(11)
      IDELTA=NDEN-NNUM
      ISTOP=11-IDELTA
```

```
      DO 10 I=1,ISTOP
   10 B(12-I)=B(12-I-IDELTA)
      DO 20 K=1,IDELTA
   20 B(K)=0.
      RETURN
      END

      SUBROUTINE PLTIMP(ICHAR)
C*
C* THIS SUBROUTINE PLOTS THE UNIT SAMPLE RESPONSE
C*
      DIMENSION PICT(1025)
      COMMON /IMPULS/ YI(1025), YS(1025), EN(1025)
      DO 5 K=1,1024
    5 EN(K+1)=K-1
      EN(1)=YI(1025)
      NFIN=YI(1025)
      DO 13 K=1,NFIN
      PICT(K+1)=YI(K)
   13 CONTINUE
      PICT(1)=YI(1025)
      CALL INIT
      CALL XFRM(2)
      CALL YFRM(2)
      IF(NFIN.LT.70)CALL LINE(-1)
      IF(NFIN.GE.70)CALL LINE(0)
      IF(NFIN.LT.70)CALL SYMBL(1)
      IF(NFIN.GE.70)CALL SYMBL(0)
      CALL SIZES(.4)
      CALL PLOT(EN,PICT)
      JFIN=YI(1025)+1
      IF(NFIN.GE.70)GO TO 40
      DO 30 K=2,JFIN
      CALL MOVEA(EN(K),PICT(K))
      CALL DRAWA(EN(K),0.0)
   30 CONTINUE
   40 CALL MOVEA(0.0,0.0)
      CALL DRAWA(EN(JFIN),0.0)
      CALL LABLE
      CALL VCURSR(ICHAR,X,Y)
      CALL FIN
      RETURN
      END

      SUBROUTINE PLTSTP(ICHAR)
C*
C* THIS SUBROUTINE PLOTS THE UNIT STEP RESPONSE
C*
```

```
      DIMENSION PICT(1025)                                               ASDF6337
      COMMON /IMPULS/ YI(1025), YS(1025), EN(1025)                       ASDF6338
      XLAST=-500.                                                        ASDF6339
      DO 5 K=1,1024                                                      ASDF6340
    5 EN(K+1)=K-1                                                        ASDF6341
      EN(1)=YS(1025)                                                     ASDF6342
      NFIN=YS(1025)                                                      ASDF6343
      DO 13 K=1,NFIN                                                     ASDF6344
      PICT(K+1)=YS(K)                                                    ASDF6345
   13 CONTINUE                                                           ASDF6346
      PICT(1)=NFIN                                                       ASDF6347
      CALL INIT                                                          ASDF6348
      CALL XFRM(2)                                                       ASDF6349
      CALL YFRM(2)                                                       ASDF6350
      IF(NFIN.LT.70)CALL LINE(-1)                                        ASDF6351
      IF(NFIN.GE.70)CALL LINE(0)                                         ASDF6352
      IF(NFIN.LT.70)CALL SYMBL(1)                                        ASDF6353
      IF(NFIN.GE.70)CALL SYMBL(0)                                        ASDF6354
      CALL SIZES(.4)                                                     ASDF6355
      CALL PLOT(EN,PICT)                                                 ASDF6356
      JFIN=NFIN+1                                                        ASDF6357
      IF(NFIN.GE.70)GO TO 40                                             ASDF6358
      DO 30 K=2,JFIN                                                     ASDF6359
      CALL MOVEA(EN(K),PICT(K))                                          ASDF6360
      CALL DRAWA(EN(K),0.)                                               ASDF6361
   30 CONTINUE                                                           ASDF6362
   40 CALL MOVEA(0.0,0.0)                                                ASDF6363
      CALL DRAWA(EN(JFIN),0.0)                                           ASDF6364
      CALL LABLE                                                         ASDF6365
      CALL VCURSR(ICHAR,X,Y)                                             ASDF6366
      CALL FIN                                                           ASDF6367
      RETURN                                                             ASDF6368
      END                                                                ASDF6369
                                                                         ASDF6370
      SUBROUTINE PLTTRF(IFNCTN,ISCALE,ICHAR)                             ASDF6371
C*                                                                       ASDF6372
C* THIS SUBROUTINE PLOTS BOTH THE PHASE AND                              ASDF6373
C* MAGNITUDE OF THE TRANSFER FUNCTION (LINEAR AND LOG)                   ASDF6374
C*                                                                       ASDF6375
      COMMON /FREQ/ YP(1026),YM(1026), PI(1026)                          ASDF6376
      DO 10 K=1,1024                                                     ASDF6377
   10 PI(K+2)=FLOAT(K-1)*3.14/1024.                                      ASDF6378
      PI(1)=1025.                                                        ASDF6379
      PI(2)=0.                                                           ASDF6380
      CALL INIT                                                          ASDF6381
      CALL XFRM(2)                                                       ASDF6382
      CALL YFRM(2)                                                       ASDF6383
      IF(IFNCTN.EQ.3)CALL PLOT(PI,YP)                                    ASDF6384
```

```
      IF(ISCALE.EQ.1)GO TO 20
      DO 15 I=3,1026
15    YM(I)=20.*ALOG10(YM(I))
      YM(2)=.01
20    IF(IFNCTN.EQ.4)CALL PLOT(PI,YM)
      CALL LABLE
      CALL TSEND
      CALL VCURSR(ICHAR,X,Y)
      CALL FIN
      RETURN
      END

      SUBROUTINE RSPNSE (IFNCTN,FLTRGN)
C*
C* THIS SUBROUTINE COMPUTES THE UNIT SAMPLE RESPONSE OF THE SYSTEM
C*
C* THE "Z" VECTOR HOLDS THE DELAYED VALUES
C* FOR THE RECURSIVE CALCULATION.
C*
      DIMENSION Z(11)
      REAL INPUT
C*
C* "ABVCTR" IS THE PAIR OF VECTORS HOLDING
C* THE VALUES OF A(K), AND B(K)
C*
      COMMON /ABVCTR/ A(11), B(11)
C*
C* "IMPULS" IS THE LABELED COMMON HOLDING THE UNIT SAMPLE RESPONSE
C*
      COMMON /IMPULS/ YI(1025), YS(1025), EN(1025)
C*
C* INITIALIZE THE Z VECTOR
C*
      XLAST=0.
      XMIN=0.
      XMAX=0.
      DELTA=0.
      DO 5 K=1,11
5     Z(K)=0.
C*
C* COMPUTE THE 1024 POINT UNIT SAMPLE OR STEP RESPONSE
C*
      DO 30 N=1,1024
C*
C* EACH VALUE OF XSUBN WILL BE ONE EXCEPT FOR THE CASE
C* OF THE UNIT SAMPLE RESPONSE WHEN N>1, AND THEN XSUBN=0.
C*
      XSUBN = 1.
```

```
      IF(N.GT.1.AND.IFNCTN.EQ.0)XSUBN = 0.
      INPUT=XSUBN
      DO 10 K=2,11
 10   INPUT=INPUT-A(K)*Z(K)
      Z(1)=INPUT
      OUTPUT=0.
      DO 20 K=1,11
 20   OUTPUT=OUTPUT+B(K)*Z(K)
      DO 25 K=1,10
 25   Z(12-K)=Z(11-K)
      IEND=N
C*
C* TEST TO SEE IF A STEADY STATE HAS BEEN ACHIEVED.
C*
 27   IF(ABS(OUTPUT).GT.XMAX)XMAX=ABS(OUTPUT)
      DIFFER=.01*XMAX
      IF(ABS(OUTPUT-XLAST).GT.DIFFER)DELTA=0.
      XLAST=OUTPUT
      DELTA=DELTA+1
      IF(DELTA.GT.20)GO TO 38
      IF(IFNCTN.EQ.0)YI(N)=OUTPUT*FLTRGN
 30   IF(IFNCTN.EQ.1)YS(N)=OUTPUT*FLTRGN
 38   IF(IFNCTN.EQ.0)YI(1025)=IEND-15
      IF(IFNCTN.EQ.1)YS(1025)=IEND-15
 999  RETURN
      END

      SUBROUTINE FRQNCY(NNUM,NDEN,FLTRGN)
C*
C* FREQNCY COMPUTES THE FREQUENCY RESPONSE FOR
C* THE FILTER BEING ANALYZED.
C*
      COMMON /ABVCTR/A(11), B(11)
      COMMON /FREQ/ YP(1026),YM(1026)
      REAL*8 X,Y,THETA,ZRO,ONE,AA(11),BB(11),XX,G(2)
      COMPLEX*16 Z, FRCTOP, FRCBOT, HOFZ, ZEXP, TERM, TERMD
      EQUIVALENCE (G,HOFZ)
      ZRO=0.0D+00
      ONE=1.0D+00
C*
C* MAKE THE COEFFICIENTS DOUBLE PRECISION
C*
      DO 20 K=1,11
      AA(K)=A(K)
      BB(K)=B(K)
 20   CONTINUE
C*
C* COMPUTE PHASE AND MAGNITUDE OF H(Z)
```

```
C*                                                                      ASDF6481
      DO 60 K=1,1025                                                    ASDF6482
      THETA=FLOAT(K-1)*3.14159265/1024.                                 ASDF6483
      Z=CDEXP(DCMPLX(ZRO,THETA))                                        ASDF6484
 25   FRCTOP=0.                                                         ASDF6485
      FRCBOT=0.                                                         ASDF6486
C*                                                                      ASDF6487
C* COMPUTE THE VALUE OF THE NUMERATOR OF H(Z)                           ASDF6488
C*                                                                      ASDF6489
      DO 40 N=1,NNUM                                                    ASDF6490
      ZEXP=Z**(N-1)                                                     ASDF6491
      TERM=DCMPLX(BB(N),ZRO)/ZEXP                                       ASDF6492
      FRCTOP=FRCTOP+TERM                                                ASDF6493
 40   CONTINUE                                                          ASDF6494
C*                                                                      ASDF6495
C* COMPUTE THE VALUE OF THE DENOMINATOR FACTORS OF H(Z)                 ASDF6496
C*                                                                      ASDF6497
      DO 50 N=2,NDEN                                                    ASDF6498
      TERMD=DCMPLX(AA(N),ZRO)/Z**(N-1)                                  ASDF6499
      FRCBOT=FRCBOT+TERMD                                               ASDF6500
 50   CONTINUE                                                          ASDF6501
C*                                                                      ASDF6502
C* GET H(Z) IN FINAL FORM                                               ASDF6503
C*                                                                      ASDF6504
      HOFZ=FRCTOP/(DCMPLX(ONE,ZRO)+FRCBOT)                              ASDF6505
C*                                                                      ASDF6506
C* COMPUTE THE MAGNITUDE OF H(Z)                                        ASDF6507
C*                                                                      ASDF6508
C*                                                                      ASDF6509
      YM(K+2)=CDABS(HOFZ)                                               ASDF6510
 55   YM(K+2)=YM(K+2)*FLTRGN                                            ASDF6511
C*                                                                      ASDF6512
C* COMPUTE THE PHASE OF H(Z)                                            ASDF6513
C*                                                                      ASDF6514
      X=G(1)                                                            ASDF6515
      Y=G(2)                                                            ASDF6516
      IF(X.EQ.ZRO.AND.Y.EQ.ZRO)YP(K+2)=YP(K+1)                          ASDF6517
      IF(X.EQ.ZRO.AND.Y.EQ.ZRO)GO TO 60                                 ASDF6518
      XX=DATAN2(Y,X)                                                    ASDF6519
      YP(K+2)=XX                                                        ASDF6520
 60   CONTINUE                                                          ASDF6521
      YP(1)=1025.                                                       ASDF6522
      YP(2)=0.                                                          ASDF6523
      YM(1)=1025.                                                       ASDF6524
      YM(2)=0.                                                          ASDF6525
      RETURN                                                            ASDF6526
      END                                                               ASDF6527
                                                                        ASDF6528
```

```
      SUBROUTINE FLTR2(FLTRGN)
C*
C* FLTR2 WRITES A LISTING OF ALL
C* POLE AND ZERO LOCATIONS FOR REVIEW BEFORE
C* PROCEEDING WITH THE PLOTTING OF RESPONSES
C*
      COMMON IROOTS(4),POLZRO(20,5)
      DIMENSION IRTTP(5)
      DATA IZ/1HZ/,IP/1HP/,IZP/2HZP/
      DATA IRTTP/2H  ,2HRZ,2HCZ,2HRP,2HCP/
      CALL INIT
      CALL FIN
      WRITE(6,102) (IZ,I=1,70)
 102  FORMAT (1X,70A1)
      WRITE (6,108)
 108  FORMAT (1X,'Z',5X,'REAL',6X,'Z',3X,'IMAGINARY',3X,'Z'
     1,6X,'RHO',6X,'Z',5X,'THETA',5X,'Z RT Z')
      WRITE (6,102) (IZ,I= 1,70)
      DO 10 I=1,10
      ITYPEP=POLZRO(I,5)+1.
      IF(POLZRO(I,5).NE.2..AND.POLZRO(I,5).NE.4.)
     1WRITE(6,104) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)
      IF(POLZRO(I,5).EQ.2..OR.POLZRO (I,5).EQ.4.)
     1WRITE(6,103) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)
 103  FORMAT(' Z   ',F10.7,'  Z +/-',F10.7,' Z   ',F10.7,
     13X,'Z +/-',F10.7,' Z ',A2,' Z')
 104  FORMAT(' Z   ',F10.7,'  Z      ',F10.7,' Z   ',F10.7,
     1'    Z     ',F10.7,' Z ',A2,' Z')
 10   CONTINUE
      WRITE(6,112) (IZP,I=1,35)
 112  FORMAT (1X,35A2)
      DO 20 I=11,20
      ITYPEP=POLZRO(I,5)+1.
      IF (POLZRO(I,5).NE.2..AND.POLZRO(I,5).NE.4.)
     1WRITE(6,106) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)
      IF(POLZRO (I,5).EQ.2..OR.POLZRO(I,5).EQ.4.)
     1WRITE(6,105) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)
 105  FORMAT(' P   ',F10.7,'  P +/-',F10.7,' P   ',F10.7,
     13X,'P +/-',F10.7,' P ',A2,' P')
 106  FORMAT(' P   ',F10.7,'  P      ',F10.7,' P   ',F10.7,
     1'    P     ',F10.7,' P ',A2,' P')
 20   CONTINUE
      WRITE(6,102) (IP, I =1,70)
      WRITE (6,107)
 107  FORMAT(//)
      WRITE(6,110) IROOTS(1)
 110  FORMAT(20X,'NUMBER OF REAL ZEROS:',I3)
      WRITE(6,111) IROOTS(2)
```

AD-A093 252 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/3

ADVANCED SIMULTION OF DIGITAL FILTERS.(U)

SEP 80 G S DOYLE

UNCLASSIFIED NL

4 OF 4

AD A093252

END

DATE FILMED

2 81

DTIC

```
111   FORMAT(20X,'NUMBER OF COMPLEX ZERO PAIRS:',I3)              ASDF6577
      WRITE(6,113) IROOTS(3)                                      ASDF6578
113   FORMAT (20X,'NUMBER OF REAL POLES:',I3)                     ASDF6579
      WRITE (6,114) IROOTS(4)                                     ASDF6580
114   FORMAT (20X, 'NUMBER CF COMPLEX POLE PAIRS:',I3)            ASDF6581
      WRITE(6,117) FLTRGN                                         ASDF6582
117   FORMAT(25X,'FILTER GAIN:',F12.8)                            ASDF6583
      WRITE(6,115)                                                ASDF6584
115   FORMAT(' HIT SPACE THEN RETURN')                            ASDF6585
      READ(5,116)                                                 ASDF6586
116   FORMAT(A1)                                                  ASDF6587
      RETURN                                                      ASDF6588
      END                                                         ASDF6589
                                                                  ASDF6590
      SUBROUTINE LABLE                                            ASDF6591
C* SUBROUTINE LABLE WRITES THE LABLES ON EACH                      ASDF6593
C* OF THE FIVE PLOTS DRAWN                                         ASDF6594
C*                                                                ASDF6595
      DIMENSION ITITLE(40)                                        ASDF6596
      CALL ANMODE                                                 ASDF6597
      READ(2,100) IX,IY,INUM,(ITITLE(J),J=1,40)                   ASDF6598
      CALL RECOVR                                                 ASDF6599
      CALL MOVABS(IX,IY)                                          ASDF6600
      CALL ANMODE                                                 ASDF6601
      WRITE(6,101) (ITITLE(J),J=1,INUM)                           ASDF6602
101   FORMAT(1X,30A1)                                             ASDF6603
      READ(2,100) IX,IY,INUM,(ITITLE(J),J=1,40)                   ASDF6604
      DO 10 I=1,INUM                                              ASDF6605
      CALL RECOVR                                                 ASDF6606
      CALL MOVABS(IX,IY)                                          ASDF6607
      CALL ANMODE                                                 ASDF6608
      WRITE(6,102) ITITLE(I)                                      ASDF6609
102   FORMAT(1X,A1)                                               ASDF6610
10    IY=IY-22                                                    ASDF6611
      READ(2,100) IX,IY,INUM,(ITITLE(J),J=1,40)                   ASDF6612
      CALL RECOVR                                                 ASDF6613
      CALL MOVABS(IX,IY)                                          ASDF6614
      CALL ANMODE                                                 ASDF6615
      WRITE(6,101) (ITITLE(J),J=1,INUM)                           ASDF6616
      READ(2,100) IX,IY,INUM,(ITITLE(J),J=1,40)                   ASDF6617
      CALL RECOVR                                                 ASDF6618
      CALL MOVABS(IX,IY)                                          ASDF6619
      CALL ANMODE                                                 ASDF6620
      WRITE(6,101) (ITITLE(J),J=1,INUM)                           ASDF6621
      CALL RECOVR                                                 ASDF6622
      CALL TSEND                                                  ASDF6623
      RETURN                                                      ASDF6624
      END                                                         ASDF6625
```

APPENDIX I: SOURCE DECK FOR ASDF COMMAND: HRD$CPY

```
C*********************************************************
C*                                                       *
C*     SOURCE DECK FOR ASDF COMMAND:  HRD$CPY           *
C*                                                       *
C*********************************************************
      COMMON IROOTS(4),POLZRO(20,5),IERROR
      COMMON /ABVCTR/ A(11), B(11)
      COMMON /IMPULS/ YI(1026), YS(1026), EN(1026)
      COMMON /FREQ/ YP(1026), YM(1026), PI(1026),YMDB(1026)
      CALL ERRSET(208,256,-1,1,1)
      LINEAR=1
      LOG=2
C*
C* INITIALIZE A AND B TO ZERO
C*
      DO 5 K=1,20
      A(K)=0.
 5    B(K)=0.
C*
C* READ THE POLZRO TABLE OFF OF THE FILE
C*
      DO 20 I=1,20
 20   READ(5,100) (POLZRO(I,J),J=1,5)
 100  FORMAT(1X,5F12.8)
C*
C* READ THE IROOTS VALUES, AND FILTER GAIN FROM THE FILE
C*
      READ(5,101) (IROOTS(I),I=1,4)
 101  FORMAT(1X,4I2)
      READ(5,102) FLTRGN
 102  FORMAT(1X,F12.8)
      FLTRGN=ABS(FLTRGN)
C*
C* PRINT OUT A TABLE OF THE POLES AND ZEROS
C*
      CALL FLTR2(FLTRGN)
C*
C* GENERATE THE A(K) COEFFICIENTS
C*
      CALL ASUBK(A,NORDER)
      NDEN=NORDER+1
C*
C* GENERATE THE B(R) COEFFICIENTS
C*
      CALL BSUBR(B,NORDER)
      NNUM=NORDER+1
C*
C* CHECK TO SEE IF THE FILTER IS CAUSAL
```

```
C*                                                                      ASDF7049
      IF(NNUM.GT.NDEN)WRITE(6,103)                                      ASDF7050
  103 FORMAT(10(/),' FILTER IS NOT CAUSAL - PROGRAM TERMINATES')        ASDF7051
      IF(NNUM.GT.NDEN)STOP                                              ASDF7052
C*                                                                      ASDF7053
C* ADJUST THE COEFFICIENTS IF NNUM DOES NOT EQUAL NDEN                  ASDF7054
C*                                                                      ASDF7055
      IF(NNUM.NE.NDEN)CALL ADJUST(NNUM,NDEN)                            ASDF7056
C*                                                                      ASDF7057
C* COMPUTE THE UNIT SAMPLE RESPONSE                                     ASDF7058
C*                                                                      ASDF7059
      CALL RSPNSE(0,FLTRGN)                                             ASDF7060
C*                                                                      ASDF7061
C* CALCULATE THE STEP RESPONSE                                          ASDF7062
C*                                                                      ASDF7063
      CALL RSPNSE(1,FLTRGN)                                             ASDF7064
C*                                                                      ASDF7065
C* CALCULATE THE FREQUENCY RESPONSE                                     ASDF7066
C*                                                                      ASDF7067
      CALL FRQNCY(NNUM,NDEN,FLTRGN)                                     ASDF7068
      CALL PRPPLT(IYIPTS,IYSPTS,NUMPLT)                                 ASDF7069
C*                                                                      ASDF7070
C* PLOT THE UNIT SAMPLE RESPONSE                                        ASDF7071
C*                                                                      ASDF7072
      CALL PLTIMP(IYIPTS,FLTRGN,NUMPLT)                                 ASDF7073
C*                                                                      ASDF7074
C* PLOT THE STEP RESPONSE                                               ASDF7075
C*                                                                      ASDF7076
      CALL PLTSTP(IYSPTS,FLTRGN,NUMPLT)                                 ASDF7077
C*                                                                      ASDF7078
C* PLOT THE PHASE OF THE TRANSFER FUNCTION                              ASDF7079
C*                                                                      ASDF7080
      CALL PLTTRF(3,LINEAR,FLTRGN,NUMPLT)                               ASDF7081
C*                                                                      ASDF7082
C* PLOT THE MAGNITUDE OF THE TRANSFER FUNCTION                          ASDF7083
C*                                                                      ASDF7084
      CALL PLTTRF(4,LINEAR,FLTRGN,NUMPLT)                               ASDF7085
      CALL PLTTRF(4,LOG,FLTRGN,NUMPLT)                                  ASDF7086
C*                                                                      ASDF7087
C* PLOT THE UNIT CIRCLE WITH POLES AND ZEROS                            ASDF7088
C*                                                                      ASDF7089
      CALL UNTCR(NUMPLT)                                                ASDF7090
C*                                                                      ASDF7091
C* OUTPUT THE DATA                                                      ASDF7092
C*                                                                      ASDF7093
      CALL OUTNUM                                                       ASDF7094
      STOP                                                              ASDF7095
      END                                                               ASDF7096
```

```
      SUBROUTINE ADJUST(NNUM,NDEN)
      COMMON /ABVCTR/A(11),B(11)
      IDELTA=NDEN-NNUM
      ISTOP=11-IDELTA
      DO 10 I=1,ISTOP
 10   B(12-I)=B(12-I-IDELTA)
      DO 20 K=1,IDELTA
 20   B(K)=0.
      NNUM=NDEN
      RETURN
      END

      SUBROUTINE OUTNUM
C*
C* OUTNUM PRINTS OUT THE NUMERICAL VALUES
C* OF THE FILTER'S RESPONSES
C*
      COMMON /IMPULS/ YI(1026), YS(1026), EN(1026)
      COMMON /FREQ/ YP(1026), YM(1026), PI(1026),YMDB(1026)
      DIMENSION M(1026)
      DO 20 L=1,1026
 20   M(L)=L-1
      DO 40 I=1,18
      CALL HEADNG
      DO 40 J=1,6
      IBEG=(I-1)*60+(J-1)*10+1
      IEND=IBEG+9
      IF(I.EQ.18.AND.J.EQ.1)IEND=1024
      WRITE(6,104)(M(K),YI(K),YS(K),YP(K),YM(K),PI(K),K=IBEG,IEND)
 104  FORMAT(10(2X,I6,2E20.8,3E21.8,/),/)
      IF(I.EQ.18.AND.J.EQ.1)GO TO 41
 40   CONTINUE
 41   WRITE(6,105)
 105  FORMAT(1H1)
      RETURN
      END

      SUBROUTINE HEADNG
C*
C* HEADNG PRINTS THE PAGE HEADINGS
C*
      DATA IX/1HX/
      WRITE(6,100)
 100  FORMAT(1H1)
      WRITE(6,101)(IX,I=1,114)
 101  FORMAT(1X,114A1)
      WRITE(6,108)
```

```
108   FORMAT(' X',112X,'X')                                               ASDF7145
      WRITE(6,103)                                                        ASDF7146
103   FORMAT(' X    N',5X,'UNIT SAMPLE RESPONSE',3X,'STEP RESPONSE',3X    ASDF7147
     X'PHASE OF H(Z) (RADIANS)',2X,'MAGNITUDE OF H(Z)',4X,                ASDF7148
     X'OMEGA (RADIANS)    X')                                             ASDF7149
      WRITE(6,108)                                                        ASDF7150
      WRITE(6,101)(IX,I=1,114)                                            ASDF7151
      WRITE(6,102)                                                        ASDF7152
102   FORMAT(/)                                                           ASDF7153
      RETURN                                                              ASDF7154
      END                                                                 ASDF7155
                                                                          ASDF7156
      SUBROUTINE UNTCR(NUMPLT)                                            ASDF7157
C*                                                                        ASDF7158
C* UNTCR PLOTS THE JNIT CIRCLE AND                                        ASDF7159
C* ALL POLES AND ZEROS                                                    ASDF7160
C*                                                                        ASDF7161
      DIMENSION UCX(362),UCY(362)                                         ASDF7162
      COMMON IROOTS(4),POLZRO(20,5),IERROR                                ASDF7163
      DO 10 I=1,361                                                       ASDF7164
      THETA=FLOAT(I-1)*2*3.14159/360.                                     ASDF7165
      UCX(I)=2.5*COS(THETA)                                               ASDF7166
      UCY(I)=2.5*SIN(THETA)                                               ASDF7167
10    CONTINUE                                                            ASDF7168
      CALL BOX(0,FLTRGN,NUMPLT)                                           ASDF7169
      CALL PLOT(.7117,.5381,-3)                                           ASDF7170
      CALL PLOT(2.75,4.25,+3)                                             ASDF7171
      CALL PLOT(8.25,4.25,+2)                                             ASDF7172
      CALL PLOT(5.5,1.5,+3)                                               ASDF7173
      CALL PLOT(5.5,7.0,+2)                                               ASDF7174
      IUP=+3                                                              ASDF7175
      IDOWN=+2                                                            ASDF7176
      HDLTA=.25                                                           ASDF7177
      VDLTA=.2                                                            ASDF7178
      X=2.75                                                              ASDF7179
      DO 70 I=1,23                                                        ASDF7180
      IF(I.NE.7.AND.I.NE.17.AND.I.NE.2.AND.I.NE.22)GO TO 71               ASDF7181
      Y=4.45                                                              ASDF7182
      CALL PLOT(X,Y,IUP)                                                  ASDF7183
      Y=4.05                                                              ASDF7184
      CALL PLOT(X,Y,IDOWN)                                                ASDF7185
      GO TO 70                                                            ASDF7186
71    Y=4.35                                                              ASDF7187
      CALL PLOT(X,Y,IUP)                                                  ASDF7188
      Y=Y-VDLTA                                                           ASDF7189
      CALL PLOT(X,Y,IDOWN)                                                ASDF7190
70    X=X+HDLTA                                                           ASDF7191
      VDLTA=.25                                                           ASDF7192
```

```
      HDLTA=.2                                                          ASDF7193
      Y=1.5                                                             ASDF7194
      DO 80 I=1,23                                                      ASDF7195
      X=5.4                                                             ASDF7196
      IF(I.NE.7.AND.I.NE.17.AND.I.NE.2.AND.I.NE.22)GO TO 81             ASDF7197
      X=5.3                                                             ASDF7198
      CALL PLOT(X,Y,IUP)                                                ASDF7199
      X=5.7                                                             ASDF7200
      CALL PLOT(X,Y,IDOWN)                                              ASDF7201
      GO TO 80                                                          ASDF7202
   81 CALL PLOT(X,Y,IUP)                                                ASDF7203
      X=X+HDLTA                                                         ASDF7204
      CALL PLOT(X,Y,IDOWN)                                              ASDF7205
   80 Y=Y+VDLTA                                                         ASDF7206
      CALL PLOT(5.5,4.25,-3)                                            ASDF7207
      CALL PLOT(UCX(1),UCY(1),+3)                                       ASDF7208
      DO 60 I=1,360                                                     ASDF7209
   60 CALL PLOT(UCX(I),UCY(I),+2)                                       ASDF7210
      CALL PLOT(UCX(1),UCY(1),+2)                                       ASDF7211
      CALL SYMBOL(-1.,+2.8125,.1875,11HUNIT CIRCLE,0.,11)               ASDF7212
      CALL SYMBOL(-2.4,-3.,.1875,26HFILTER POLE ZERO LOCATIONS,0.,26)   ASDF7213
      DO 30 I=1,20                                                      ASDF7214
      DO 30 J=1,3                                                       ASDF7215
   30 POLZRO(I,J)=POLZRO(I,J)*2.5                                       ASDF7216
      DO 40 I=1,10                                                      ASDF7217
      IF(POLZRO(I,3).EQ.-25.)GO TO 40                                   ASDF7218
      IF(POLZRO(I,3).LE.2.5)GO TO 45                                    ASDF7219
      POLZRO(I,1)=2.625*COS(POLZRO(I,4))                                ASDF7220
      POLZRO(I,2)=2.625*SIN(POLZRO(I,4))                                ASDF7221
      CALL SYMBOL(POLZRO(I,1),POLZRO(I,2),.25,11,0.,-1)                 ASDF7222
      IF(POLZRO(I,5).NE.1.)CALL SYMBOL(POLZRO(I,1),-POLZRO(I,2),.25,    ASDF7223
     X11,0.,-1)                                                         ASDF7224
      GO TO 40                                                          ASDF7225
   45 CALL SYMBOL(POLZRO(I,1),POLZRO(I,2),.25,1,0.,-1)                  ASDF7226
      IF(POLZRO(I,5).NE.1.)CALL SYMBOL (POLZRO(I,1),-POLZRO(I,2),.25,   ASDF7227
     X1,0.,-1)                                                          ASDF7228
   40 CONTINUE                                                          ASDF7229
      DO 50 I=11,20                                                     ASDF7230
      IF(POLZRO(I,1).EQ.-25.)GO TO 50                                   ASDF7231
      CALL SYMBOL(POLZRO(I,1),POLZRO(I,2),.25,4,0.,-1)                  ASDF7232
      IF(POLZRO(I,5).NE.3)CALL SYMBOL(POLZRO(I,1),-POLZRO(I,2),.25,4,0. ASDF7233
     X,-1)                                                              ASDF7234
   50 CONTINUE                                                          ASDF7235
      CALL PLOT (0.,0.,+999)                                            ASDF7236
      RETURN                                                            ASDF7237
      END                                                               ASDF7238
                                                                        ASDF7239
      SUBROUTINE ASJBK (POLE,NORDER)                                    ASDF7240
```

```
C*                                                                      ASDF7241
C* GENERATE THE AK COEFFICIENTS FROM THE POLES                          ASDF7242
C*                                                                      ASDF7243
      COMMON IROOTS(4), POLZRO(20,5), IERROR                            ASDF7244
      COMPLEX POLES(11), ACOEF(11)                                      ASDF7245
      REAL POLE(11)                                                     ASDF7246
C*                                                                      ASDF7247
C* INITIALIZE ALL THE VECTORS AND INDICES TO ZERO                       ASDF7248
C*                                                                      ASDF7249
      DO 5 K=1,11                                                       ASDF7250
      POLES(K)=(0.,0.)                                                  ASDF7251
      POLE(K)=0.                                                        ASDF7252
 5    ACOEF(K)=0.                                                       ASDF7253
C*                                                                      ASDF7254
C* "J" INDEXES THE TABLE "POLES"                                        ASDF7255
C* "K" INDEXES THE TABLE "POLZRO"                                       ASDF7256
C*                                                                      ASDF7257
      J=1                                                               ASDF7258
      K=1                                                               ASDF7259
C*                                                                      ASDF7260
C* COMPUTE THE ORDER OF THE SYSTEM OF POLES                             ASDF7261
C*                                                                      ASDF7262
      NORDER=IROOTS(3)+2*IROOTS(4)                                      ASDF7263
      IF(NORDER.NE.0)GO TO 10                                           ASDF7264
      POLE(1)=1.0                                                       ASDF7265
      RETURN                                                            ASDF7266
C*                                                                      ASDF7267
C* THE VECTOR "POLES" WILL CONTAIN A LIST OF ALL                        ASDF7268
C* THE POLES, IE., X+J0, X+JY, X-JY, . . .                              ASDF7269
C*                                                                      ASDF7270
 10   POLES(J)=CMPLX(POLZRO(10+K,1),POLZRO(10+K,2))                     ASDF7271
C*                                                                      ASDF7272
C* TEST TO SEE IF ALL ROOTS HAVE BEEN                                   ASDF7273
      IF(J.EQ.NORDER)GO TO 20                                           ASDF7276
      J=J+1                                                             ASDF7277
C*                                                                      ASDF7275
C*                                                                      ASDF727
C* IF THE ROOT IS REAL CONTINUE PROCESING THE NEXT ROOT                 ASDF7279
C*                                                                      ASDF7280
      IF(POLZRO(K+10,5).NE.3)GO TO 15                                   ASDF7281
      K=K+1                                                             ASDF7282
      GO TO 10                                                          ASDF7283
C*                                                                      ASDF7284
C* SINCE THE ROOT IS COMPLEX PUT THE CONJUGATE                          ASDF7285
C* OF THE ROOT IN THE "POLES" TABLE.                                    ASDF7286
C*                                                                      ASDF7287
 15   POLES(J)=CMPLX(POLZRO(10+K,1),-POLZRO(10+K,2))                    ASDF7288
      IF(J.EQ.NORDER) GO TO 20                                          ASDF7289
```

```
      J=J+1                                                             ASDF7290
      K=K+1                                                             ASDF7291
      GO TO 10                                                          ASDF7292
C*                                                                      ASDF7293
C* GENERATE THE COEFFICIENTS FOR THE D(Z) POLYNOMIAL                    ASDF7294
C* OF THE TRANSFER FUNCTION H(Z).                                       ASDF7295
C*                                                                      ASDF7296
 20   CALL COEFF(POLES,NORDER,ACOEF)                                    ASDF7297
C*                                                                      ASDF7298
C* CHANGE THE COEFFICIENTS FROM COMPLEX ASUBK TO REAL ASUBK             ASDF7299
C*                                                                      ASDF7300
      DO 40 J=1,11                                                      ASDF7301
 40   POLE(J)=REAL(ACOEF(J))                                            ASDF7302
      RETURN                                                            ASDF7303
      END                                                               ASDF7304
                                                                        ASDF7305
      SUBROUTINE BSUBR(ZERO,NORDER)                                     ASDF7306
C*                                                                      ASDF7307
C* GENERATE THE B(R) COEFFICIENTS FROM THE ZEROS                        ASDF7308
C*                                                                      ASDF7309
      COMMON IROOTS(4),POLZRO(20,5),IERROR                              ASDF7310
      COMPLEX ZEROS(11), BCOEF(11)                                      ASDF7311
      REAL ZERO(11)                                                     ASDF7312
C*                                                                      ASDF7313
C* INITIALIZE ALL VECTORS AND INDICES                                   ASDF7314
C*                                                                      ASDF7315
      DO 5 K=1,11                                                       ASDF7316
      ZEROS(K)=(0.,0.)                                                  ASDF7317
      ZERO(K)=0.                                                        ASDF7318
 5    BCOEF(K)=(0.,0.)                                                  ASDF7319
C*                                                                      ASDF7320
C* "J" INDEXES THE "ZEROS" TABLE                                        ASDF7321
C* "K" INDEXES THE "POLZRO" TABLE                                       ASDF7322
C*                                                                      ASDF7323
      J=1                                                               ASDF7324
      K=1                                                               ASDF7325
C*                                                                      ASDF7326
C* COMPUTE THE ORDER OF THE SYSTEM OF ZEROS.                            ASDF7327
C*                                                                      ASDF7328
      NORDER=IROOTS(1)+2*IROOTS(2)                                      ASDF7329
      IF(NORDER.NE.0)GO TO 10                                           ASDF7330
      ZERO(1)=1.0                                                       ASDF7331
      RETURN                                                            ASDF7332
C*                                                                      ASDF7333
C* THE VECTOR "ZEROS" CONTAINS A LIST OF ALL                            ASDF7334
C* OF THE ZEROS, IE., X+J0, X+JY, X-JY, . . .                           ASDF7335
C*                                                                      ASDF7336
 10   ZEROS(J)=CMPLX(POLZRO(K,1),POLZRO(K,2))                           ASDF7337
```

```
C*                                                                      ASDF7338
C* CHECK TO SEE IF ALL THE ROOTS HAVE BEEN PROCESSED                    ASDF7339
C*                                                                      ASDF7340
      IF(J.EQ.NORDER)GO TO 20                                           ASDF7341
      J=J+1                                                             ASDF7342
C*                                                                      ASDF7343
C* IF THE ROOT IS REAL CONTINUE PROCESSING THE NEXT ROOT                 ASDF7344
C*                                                                      ASDF7345
      IF(POLZRO(K,5).NE.1)GO TO 15                                      ASDF7346
      K=K+1                                                             ASDF7347
      GO TO 10                                                          ASDF7348
C*                                                                      ASDF7349
C* SINCE THE ROOT IS COMPLEX, ENTER THE CONJUGATE                        ASDF7350
C* INTO THE TABLE "ZEROS".                                              ASDF7351
C*                                                                      ASDF7352
 15   ZEROS(J)=CMPLX(POLZRO(K,1),-POLZRO(K,2))                          ASDF7353
      IF(J.EQ.NORDER)GO TO 20                                           ASDF7354
      J=J+1                                                             ASDF7355
      K=K+1                                                             ASDF7356
      GO TO 10                                                          ASDF7357
C*                                                                      ASDF7358
C* GENERATE THE BSUBR COEFFICIENTS                                      ASDF7359
C*                                                                      ASDF7360
 20   CALL COEFF(ZEROS, NORDER, BCOEF)                                  ASDF7361
C*                                                                      ASDF7362
C* MAKE THE COEFFICIENTS BSUBR REAL.                                    ASDF7363
C*                                                                      ASDF7364
      DO 40 J=1,11                                                      ASDF7365
 40   ZERO(J)=REAL(BCOEF(J))                                            ASDF7366
      RETURN                                                            ASDF7367
      END                                                               ASDF7368
                                                                        ASDF7369
      SUBROUTINE COEFF(V,N,H1)                                          ASDF7370
C*                                                                      ASDF7371
C* "V" - VECTOR OF ROOTS; "N" - THE ORDER OF THE SYSTEM                  ASDF7372
C* "H1" - VECTOR WITH THE FINAL COEFFICIENTS                             ASDF7373
C* H (1)=Z**(0), H1(2)=Z**(-1), . . . ,H1(11)=Z**(-10)                   ASDF7374
C*                                                                      ASDF7375
      COMPLEX H1(11),H2(11),H3(11),V(11)                                ASDF7376
C* INITIALIZE VECTORS                                                   ASDF7377
      IZERO=0                                                           ASDF7378
      DO 10 I=1,11                                                      ASDF7379
      H1(I)=(0.,0.)                                                     ASDF7380
      H2(I)=(0.,0.)                                                     ASDF7381
 10   H3(I)=(0.,0.)                                                     ASDF7382
      H1(1)=(1.,0.)                                                     ASDF7383
C*                                                                      ASDF7384
C* THE ROOTS COME FROM THE FORMER SUBROUTINE                             ASDF7385
```

```
C* THESE ARE ROOTS OF THE POLYNOMIALS N(Z), AND D(Z)                    ASDF7386
C* WHERE H(Z)=N(Z)/D(Z).                                                ASDF7387
C* THE FACTORS OF THE POLYNOMIAL ARE OF THE FORM Z-ROOT                 ASDF7388
C*                                                                      ASDF7389
      DO 15 K=1,11                                                      ASDF7390
   15 V(K)=-V(K)                                                        ASDF7391
C*                                                                      ASDF7392
C* WHEN K=N, WE HAVE GENERATED THE H1 COEFFICIENTS                       ASDF7393
C* OF N(Z), OR D(Z) OF H(Z)=N(Z)/D(Z)                                   ASDF7394
C* THE REMAINING ITERATIONS ARE PERFORMED TO SIMPLIFY                    ASDF7395
C* OBTAINING COEFFICIENTS FOR N(Z**-1), AND D(Z**-1) LATER               ASDF7396
C*                                                                      ASDF7397
      DO 30 K=1,10                                                      ASDF7398
C*                                                                      ASDF7399
C* "H2" REPRESENTS THE PREVIOUS RESULT TIMES Z**(-1)                     ASDF7400
C*                                                                      ASDF7401
      DO 40 I=1,10                                                      ASDF7402
   40 H2(I+1)=H1(I)                                                     ASDF7403
      H2(1)=(0.,0.)                                                     ASDF7404
      DO 50 I=1,11                                                      ASDF7405
   50 H3(I)=H1(I)*V(1)                                                  ASDF7406
C*                                                                      ASDF7407
C* GENERATE THE NEW FINAL PRODUCT WITH K ROOTS MULTIPLIED.               ASDF7408
C*                                                                      ASDF7409
      DO 60 I=1,11                                                      ASDF7410
   60 H1(I)=H2(I)+H3(I)                                                 ASDF7411
C*                                                                      ASDF7412
C* GET THE NEXT ROOT TO BE MULTIPLIED AND PUT IT INTO "V(1)"             ASDF7413
C*                                                                      ASDF7414
   65 DO 70 I=1,10                                                      ASDF7415
   70 V(I)=V(I+1)                                                       ASDF7416
   30 CONTINUE                                                          ASDF7417
C*                                                                      ASDF7418
C* TO GET THE COEFFICIENTS FOR H(Z**-1)=N(Z**-1)/D(Z**-1)                ASDF7419
C* THE ORDER OF THE COEFFICIENTS MUST BE REVERSED.                       ASDF7420
C*                                                                      ASDF7421
      DO 80 I=1,11                                                      ASDF7422
   80 H2(I)=H1(12-I)                                                    ASDF7423
      DO 90 I=1,11                                                      ASDF7424
   90 H1(I)=H2(I)                                                       ASDF7425
   95 RETURN                                                            ASDF7426
      END                                                               ASDF7427
                                                                        ASDF7428
      SUBROUTINE RSPNSE (IFNCTN,FLTRGN)                                 'SDF7429
C*                                                                      ASDF7430
C* THIS SUBROUTINE COMPUTES THE UNIT SAMPLE RESPONSE OF THE SYSTEM       ASDF7431
C*                                                                      ASDF7432
C* THE "Z" VECTOR HOLDS THE DELAYED VALUES                               ASDF7433
```

```
C* FOR THE RECURSIVE CALCULATION.                                       ASDF7434
C*                                                                      ASDF7435
      REAL*8 XMIN,XMAX,XLAST,XSUBN,INPUT,OUTPUT,AA(11),BB(11),          ASDF7436
     XZ(11),DIFFER,DFTRGN,ZRO,ONE,PTOONE                                ASDF7437
C*                                                                      ASDF7438
C* "ABVCTR" IS THE PAIR OF VECTORS HOLDING                              ASDF7439
C* THE VALUES OF A(K), AND B(K)                                         ASDF7440
C*                                                                      ASDF7441
      COMMON /ABVCTR/ A(11), B(11)                                      ASDF7442
C*                                                                      ASDF7443
C* "IMPULS" IS THE LABELED COMMON HOLDING THE UNIT SAMPLE RESPONSE      ASDF7444
C*                                                                      ASDF7445
      COMMON /IMPULS/ YI(1026), YS(1026), EN(1026)                      ASDF7446
C*                                                                      ASDF7447
C* INITIALIZE THE CONSTANTS AND Z VECTOR                                ASDF7448
C*                                                                      ASDF7449
      IFLAG=0                                                           ASDF7450
      ZRO=0.0D+00                                                       ASDF7451
      ONE=1.0D+00                                                       ASDF7452
      PTOONE=1.0D-02                                                    ASDF7453
      XMIN=ZRO                                                          ASDF7454
      XMAX=ZRO                                                          ASDF7455
      XLAST=ZRO                                                         ASDF7456
      DELTA=0.                                                          ASDF7457
      DFTRGN=FLTRGN                                                     ASDF7458
      DO 4 K=1,11                                                       ASDF7459
      AA(K)=A(K)                                                        ASDF7460
    4 BB(K)=B(K)                                                        ASDF7461
      DO 5 K=1,11                                                       ASDF7462
    5 Z(K)=0.                                                           ASDF7463
C*                                                                      ASDF7464
C* COMPUTE THE 1024 POINT UNIT SAMPLE OR STEP RESPONSE                  ASDF7465
C*                                                                      ASDF7466
      DO 30 N=1,1024                                                    ASDF7467
C*                                                                      ASDF7468
C* EACH VALUE OF XSUBN WILL BE ONE EXCEPT FOR THE CASE                  ASDF7469
C* OF THE UNIT SAMPLE RESPONSE WHEN N>1, AND THEN XSUBN=0.              ASDF7470
C*                                                                      ASDF7471
      XSUBN=ONE                                                         ASDF7472
      IF(N.GT.1.AND.IFNCTN.EQ.0)XSUBN=ZRO                               ASDF7473
      INPUT=XSUBN                                                       ASDF7474
      DO 10 K=2,11                                                      ASDF7475
   10 INPUT=INPUT-A(K)*Z(K)                                             ASDF7476
      Z(1)=INPUT                                                        ASDF7477
      OUTPUT=ZRO                                                        ASDF7478
      DO 20 K=1,11                                                      ASDF7479
   20 OUTPUT=OUTPUT+B(K)*Z(K)                                           ASDF7480
      DO 25 K=1,10                                                      ASDF7481
```

```
 25    Z(12-K)=Z(11-K)                                                  ASDF7482
       IEND=N                                                           ASDF7483
C*                                                                      ASDF7484
C* TEST TO SEE IF A STEADY STATE HAS BEEN ACHIEVED.                     ASDF7485
C*                                                                      ASDF7486
       IF(DABS(OUTPUT).GT.XMAX)XMAX=DABS(OUTPUT)                        ASDF7487
       DIFFER=PTOONE*XMAX                                               ASDF7488
       IF(DABS(OUTPUT-XLAST).GT.DIFFER)DELTA=0.                         ASDF7489
       XLAST=OUTPUT                                                     ASDF7490
       DELTA=DELTA+1                                                    ASDF7491
       IF(IFNCTN.EQ.0)YI(N)=OUTPUT*DFTRGN                               ASDF7492
       IF(IFNCTN.EQ.1)YS(N)=OUTPUT*DFTRGN                               ASDF7493
       IF(DELTA.LE.20.)GO TO 30                                         ASDF7494
       IFLAG=1                                                          ASDF7495
       DELTA=-2000.                                                     ASDF7496
 38    IF(IFNCTN.EQ.0)YI(1025)=IEND-15                                  ASDF7497
       IF(IFNCTN.EQ.1)YS(1025)=IEND-15                                  ASDF7498
 30    CONTINUE                                                         ASDF7499
       IF(IFLAG.EQ.0.AND.IFNCTN.EQ.0)YI(1025)=1024.                     ASDF7500
       IF(IFLAG.EQ.0.AND.IFNCTN.EQ.1)YS(1025)=1024.                     ASDF7501
 999   RETURN                                                           ASDF7502
       END                                                              ASDF7503
                                                                        ASDF7504
       SUBROUTINE FRQNCY(NNUM,NDEN,FLTRGN)                              ASDF7505
       COMMON /ABVCTR/A(11), B(11)                                      ASDF7506
       COMMON /FREQ/ YP(1026),YM(1026),PI(1026),YMDB(1026)              ASDF7507
       REAL*8 X,Y,THETA,ZRO,ONE,AA(11),BB(11),XX,G(2)                   ASDF7508
       COMPLEX*16 Z, FRCTOP, FRCBOT, HOFZ, ZEXP, TERM, TERMD            ASDF7509
       EQUIVALENCE (G,HOFZ)                                             ASDF7510
       ZRO=0.0D+00                                                      ASDF7511
       ONE=1.0D+00                                                      ASDF7512
C*                                                                      ASDF7513
C* MAKE THE COEFFICIENTS DOUBLE PRECISION                               ASDF7514
C*                                                                      ASDF7515
       DO 456 KKK=1,11                                                  ASDF7516
       AA(KKK)=A(KKK)                                                   ASDF7517
       BB(KKK)=B(KKK)                                                   ASDF7518
 456   CONTINUE                                                         ASDF7519
       DO 60 K=1,1025                                                   ASDF7520
       THETA=FLOAT(K-1)*3.14159265/1024.                                ASDF7521
       Z=CDEXP(DCMPLX(ZRO,THETA))                                       ASDF7522
 25    FRCTOP=0.                                                        ASDF7523
       FRCBOT=0.                                                        ASDF7524
       DO 40 N=1,NNUM                                                   ASDF7525
       ZEXP=Z**(N-1)                                                    ASDF7526
       TERM=DCMPLX(BB(N),ZRO)/ZEXP                                      ASDF7527
       FRCTOP=FRCTOP+TERM                                               ASDF7528
 40    CONTINUE                                                         ASDF7529
```

```
      DO 50 N=2,NDEN                                                    ASDF7530
      TERMD=DCMPLX(AA(N),ZRO)/Z**(N-1)                                  ASDF7531
      FRCBOT=FRCBOT+TERMD                                               ASDF7532
   50 CONTINUE                                                          ASDF7533
C*                                                                      ASDF7534
C* CALCULATE H(Z)                                                       ASDF7535
C*                                                                      ASDF7536
      HOFZ=FRCTOP/(DCMPLX(ONE,ZRO)+FRCBOT)                              ASDF7537
      YM(K)=CDABS(HOFZ)                                                 ASDF7538
   55 YM(K)=YM(K)*FLTRGN                                                ASDF7539
      X=G(1)                                                            ASDF7540
      Y=G(2)                                                            ASDF7541
      IF(X.EQ.ZRO.AND.Y.EQ.ZRO)YP(K+1)=YP(K)                            ASDF7542
      IF(X.EQ.ZRO.AND.Y.EQ.ZRO)GO TO 60                                 ASDF7543
      XX=DATAN2(Y,X)                                                    ASDF7544
      YP(K)=XX                                                          ASDF7545
   60 CONTINUE                                                          ASDF7546
      RETURN                                                            ASDF7547
      END                                                               ASDF7548
                                                                        ASDF7549
      SUBROUTINE FLTR2(FLTRGN)                                          ASDF7550
      COMMON IROOTS(4),POLZRO(20,5)                                     ASDF7551
      DIMENSION IRTTP(5)                                                ASDF7552
      DATA IZ/1HZ/,IP/1HP/,IZP/2HZP/                                    ASDF7553
      DATA IRTTP/2H  ,2HRZ,2HCZ,2HRP,2HCP/                              ASDF7554
      WRITE(6,100)                                                      ASDF7555
  100 FORMAT(1H1)                                                       ASDF7556
      WRITE(6,101)                                                      ASDF7557
  101 FORMAT(///////////////////////)                                   ASDF7558
      WRITE(6,102) (IZ,I=1,70)                                          ASDF7559
  102 FORMAT (31X,70A1)                                                 ASDF7560
      WRITE (6,108)                                                     ASDF7561
  108 FORMAT (31X,'Z',5X,'REAL',6X,'Z',3X,'IMAGINARY',3X,'Z'            ASDF7562
     1,6X,'RHO',6X,'Z',5X,'THETA',5X,'Z RT Z')                          ASDF7563
      WRITE (6,102) (IZ,I= 1,70)                                        ASDF7564
      DO 10 I=1,10                                                      ASDF7565
      ITYPEP=POLZRO(I,5)+1.                                             ASDF7566
      IF(POLZRO(I,5).NE.2..AND.POLZRO(I,5).NE.4.)                       ASDF7567
     1WRITE(6,104) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)                  ASDF7568
      IF(POLZRO(I,5).EQ.2..OR.POLZRO (I,5).EQ.4.)                       ASDF7569
     1WRITE(6,103) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)                  ASDF7570
  103 FORMAT(30X,' Z   ',F10.7,'  Z +/-',F10.7,' Z  ',F10.7,            ASDF7571
     13X,'Z +/-',F10.7,' Z ',A2,' Z')                                   ASDF7572
  104 FORMAT(30X,' Z   ',F10.7,'  Z     ',F10.7,' Z  ',F10.7,           ASDF7573
     1'   Z    ',F10.7,' Z ',A2,' Z')                                   ASDF7574
   10  CONTINUE                                                         ASDF7575
      WRITE(6,112) (IZP,I=1,35)                                         ASDF7576
  112 FORMAT (31X,35A2)                                                 ASDF7577
```

```
      DO 20 I=11,20                                                     ASDF7578
      ITYPEP=POLZRO(I,5)+1.                                             ASDF7579
      IF (POLZRO(I,5).NE.2..AND.POLZRO(I,5).NE.4.)                      ASDF7580
     1WRITE(6,106) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)                  ASDF7581
      IF(POLZRO (I,5).EQ.2..OR.POLZRO(I,5).EQ.4.)                       ASDF7582
     1WRITE(6,105) (POLZRO(I,J), J=1,4), IRTTP(ITYPEP)                  ASDF7583
 105  FORMAT(30X' P    ',F10.7,'   P +/-',F10.7,' P   ',F10.7,          ASDF7584
     13X,'P +/-',F10.7,' P ',A2,' P')                                   ASDF7585
 106  FORMAT(30X' P    ',F10.7,'   P      ',F10.7,' P   ',F10.7,        ASDF7586
     1'    P     ',F10.7,' P ',A2,' P')                                 ASDF7587
 20    CONTINUE                                                         ASDF7588
      WRITE(6,102) (IP, I =1,70)                                        ASDF7589
      WRITE (6,107)                                                     ASDF7590
 107  FORMAT(//)                                                        ASDF7591
      WRITE(6,110) IROOTS(1)                                            ASDF7592
 110  FORMAT(50X,'NUMBER OF REAL ZEROS:',I3)                            ASDF7593
      WRITE(6,111) IROOTS(2)                                            ASDF7594
 111  FORMAT(50X,'NUMBER OF COMPLEX ZERO PAIRS:',I3)                    ASDF7595
      WRITE(6,113) IROOTS(3)                                            ASDF7596
 113  FORMAT (50X,'NUMBER OF REAL POLES:',I3)                           ASDF7597
      WRITE (6,114) IROOTS(4)                                           ASDF7598
 114  FORMAT (50X, 'NUMBER OF COMPLEX POLE PAIRS:',I3)                  ASDF7599
      WRITE(6,115) FLTRGN                                               ASDF7600
 115  FORMAT(50X,'FILTER GAIN:',F12.8)                                  ASDF7601
      WRITE (6,100)                                                     ASDF7602
      RETURN                                                            ASDF7603
      END                                                               ASDF7604
                                                                        ASDF7605
      SUBROUTINE PRPPLT(IYIPTS,IYSPTS,NUMPLT)                           ASDF7606
C*                                                                      ASDF7607
C* THIS SUBROUTINE SCALES THE ORDINATE                                  ASDF7608
C* DATA BEFORE PLOTTING                                                 ASDF7609
C*                                                                      ASDF7610
      COMMON /IMPULS/YI(1026),YS(1026),EN(1026)                         ASDF7611
      COMMON /FREQ/YP(1026), YM(1026),PI(1026),YMDB(1026)               ASDF7612
      NUMPLT=1                                                          ASDF7613
      IYIPTS=YI(1025)                                                   ASDF7614
      IYSPTS=YS(1025)                                                   ASDF7615
      CALL WINDOW(0.,45.,0.,21.11)                                      ASDF7616
      CALL PLOTS(0.,0.,0.)                                              ASDF7617
      ISUB=2                                                            ASDF7618
      IF(ABS(YI(1)).LE.ABS(YI(2)))ISUB=1                                ASDF7619
      TEMP=YI(ISUB)                                                     ASDF7620
      YI(ISUB)=0.                                                       ASDF7621
      CALL SCALE(YI,5.,1024,+1)                                         ASDF7622
      YI(ISUB)=TEMP                                                     ASDF7623
      ISUB=2                                                            ASDF7624
      IF(ABS(YS(1)).LE.ABS(YS(2)))ISUB=1                                ASDF7625
```

```
      TEMP=YS(ISUB)                                                     ASDF7626
      YS(ISUB)=0.                                                       ASDF7627
      CALL SCALE(YS,5.,1024,+1)                                         ASDF7628
      YS(ISUB)=TEMP                                                     ASDF7629
      TEMP=YM(1)                                                        ASDF7630
      YM(1)=0.                                                          ASDF7631
      CALL SCALE(YM,5.,1024,+1)                                         ASDF7632
      YM(1)=TEMP                                                        ASDF7633
      DO 10 I=1,1024                                                    ASDF7634
      IF(YM(I).EQ.0.)YM(I)=1E-70                                        ASDF7635
   10 YMDB(I)=20.*ALOG10(YM(I))                                         ASDF7636
      ISUB=2                                                            ASDF7637
      IF(ABS(YMDB(1)).LE.ABS(YMDB(2)))ISUB=1                            ASDF7638
      TEMP=YMDB(ISUB)                                                   ASDF7639
C*                                                                      ASDF7640
C* THIS STATEMENT PREVENTS THE STAIRCASE FOR                            ASDF7641
C* ALL PASS FILTERS                                                     ASDF7642
C*                                                                      ASDF7643
      YMDB(ISUB)=-10.                                                   ASDF7644
      CALL SCALE(YMDB,5.,1024,+1)                                       ASDF7645
      YMDB(ISUB)=TEMP                                                   ASDF7646
   20 CALL SCALE(YP,5.,1024,+1)                                         ASDF7647
      RETURN                                                            ASDF7648
      END                                                               ASDF7649
                                                                        ASDF7650
      SUBROUTINE PLTIMP(LAST,FLTRGN,NUMPLT)                             ASDF7651
C*                                                                      ASDF7652
C* THIS SUBROUTINE PLOTS THE UNIT SAMPLE RESPONSE                       ASDF7653
C*                                                                      ASDF7654
      COMMON /IMPULS/YI(1026),YS(1026),EN(1026)                         ASDF7655
      CALL BOX(1,FLTRGN,NUMPLT)                                         ASDF7656
      DO 5 I=1,1024                                                     ASDF7657
    5 EN(I)=I-1                                                         ASDF7658
      CALL SCALE(EN,8.,LAST,+1)                                         ASDF7659
      CALL AXIS(2.179,2.2012,1HN,-1,8.,0.,EN(LAST+1),EN(LAST+2))        ASDF7660
      CALL AXIS(2.179,2.2012,20HUNIT SAMPLE RESPONSE,+20,5.,90.,YI(1025)ASDF7661
     X,YI(1026))                                                        ASDF7662
      CALL PLOT(2.179,2.2012,-3)                                        ASDF7663
      YZERO=(-YI(1025)/YI(1026))                                        ASDF7664
      ISTOP=8.*EN(LAST+2)+1.5                                           ASDF7665
      IF(ISTOP.GT.1024)ISTOP=1024                                       ASDF7666
      FVAL=EN(LAST+1)                                                   ASDF7667
      DELTA=EN(LAST+2)                                                  ASDF7668
      DO 7 I=1,ISTOP                                                    ASDF7669
    7 EN(I)=I-1                                                         ASDF7670
      DO 10 K=1,ISTOP                                                   ASDF7671
      X=(EN(K)-FVAL)/DELTA                                              ASDF7672
      Y=(YI(K)-YI(1025))/YI(1026)                                       ASDF7673
```

```
      IF(Y.GE.YZERO.AND.ISTOP.LE.80)CALL SYMBOL(X,Y,.1,1,180.,-1)       ASDF7674
      IF(Y.LT.YZERO.AND.ISTOP.LE.80)CALL SYMBOL(X,Y,.1,1,0.,-1)         ASDF7675
      CALL PLOT(X,Y,+3)                                                  ASDF7676
      IF(YZERO.LT.0.)CALL PLOT(X,0.,+2)                                  ASDF7677
 10   IF(YZERO.GE.0.)CALL PLOT(X,YZERO,+2)                               ASDF7678
      IF(YZERO.LT.0)GO TO 999                                            ASDF7679
      CALL PLOT(0.,YZERO,+3)                                             ASDF7680
      CALL PLOT(X,YZERO,+2)                                              ASDF7681
 999  CALL PLOT(-2.179,-2.2012,-3)                                       ASDF7682
      NUMPLT=NUMPLT+1                                                    ASDF7683
      RETURN                                                             ASDF7684
      END                                                                ASDF7685
                                                                         ASDF7686
      SUBROUTINE PLTSTP(LAST,FLTRGN,NUMPLT)                              ASDF7687
C*                                                                       ASDF7688
C* THIS SUBROUTINE PLOTS THE UNIT STEP RESPONSE                          ASDF7689
C*                                                                       ASDF7690
      COMMON /IMPULS/YI(1026),YS(1026),EN(1026)                          ASDF7691
      CALL BOX(1,FLTRGN,NUMPLT)                                          ASDF7692
      DO 5 I=1,1024                                                      ASDF7693
 5    EN(I)=I-1                                                          ASDF7694
      CALL SCALE(EN,8.,LAST,+1)                                          ASDF7695
      CALL AXIS(2.179,2.2012,13HSTEP RESPONSE,+13,5.,90.,YS(1025),      ASDF7696
     XYS(1026))                                                          ASDF7697
      CALL AXIS(2.179,2.2012,1HN,-1,8.,0.,EN(LAST+1),EN(LAST+2))         ASDF7698
      CALL PLOT(2.179,2.2012,-3)                                         ASDF7699
      YZERO=(-YS(1025)/YS(1026))                                         ASDF7700
      ISTOP=8.*EN(LAST+2)+1.5                                            ASDF7701
      IF(ISTOP.GT.1024)ISTOP=1024                                        ASDF7702
      FVAL=EN(LAST+1)                                                    ASDF7703
      DELTA=EN(LAST+2)                                                   ASDF7704
      DO 7 I=1,ISTOP                                                     ASDF7705
 7    EN(I)=I-1                                                          ASDF7706
      DO 10 K=1,ISTOP                                                    ASDF7707
      X=(EN(K)-FVAL)/DELTA                                               ASDF7708
      Y=(YS(K)-YS(1025))/YS(1026)                                        ASDF7709
      IF(ISTOP.LE.80)CALL SYMBOL(X,Y,.1,1,180.,-1)                       ASDF7710
      CALL PLOT(X,Y,+3)                                                  ASDF7711
      IF(YZERO.LT.0.)CALL PLOT(X,0.,+2)                                  ASDF7712
 10   IF(YZERO.GE.0.)CALL PLOT(X,YZERO,+2)                               ASDF7713
      IF(YZERO.LT.0)GO TO 999                                            ASDF7714
      CALL PLOT(0.,YZERO,+3)                                             ASDF7715
      CALL PLOT(X,YZERO,+2)                                              ASDF7716
 999  CALL PLOT(-2.179,-2.2012,-3)                                       ASDF7717
      NUMPLT=NUMPLT+1                                                    ASDF7718
      RETURN                                                             ASDF7719
      END                                                                ASDF7720
                                                                         ASDF7721
```

```
      SUBROUTINE PLTTRF(IFNCTN,ITYPE,FLTRGN,NUMPLT)                     ASDF7722
C*                                                                      ASDF7723
C* THIS SUBROUTINE PLOTS THE PHASE AND                                  ASDF7724
C* MAGNITUDE OF THE TRANSFER FUNCTION                                   ASDF7725
C*                                                                      ASDF7726
      COMMON /FREQ/YP(1026),YM(1026),PI(1026),YMDB(1026)               ASDF7727
      DO 10 K=1,1024                                                    ASDF7728
   10 PI(K)=FLOAT(K-1)*3.14159/1024.                                    ASDF7729
      CALL BOX(1,FLTRGN,NUMPLT)                                         ASDF7730
      CALL SCALE(PI,8.,1024,+1)                                         ASDF7731
      CALL AXIS(2.179,2.2012,7HRADIANS,-7,8.,0.,PI(1025),PI(1026))      ASDF7732
      IF(IFNCTN.EQ.4)GO TO 30                                           ASDF7733
      CALL SCALE(YP,5.,1024,+1)                                         ASDF7734
      CALL AXIS(2.179,2.2012,24HPHASE OF H(Z)  (RADIANS),+24,5.,90.,   ASDF7735
     XYP(1025),YP(1026))                                                ASDF7736
      CALL PLOT(2.179,2.2012,-3)                                        ASDF7737
      CALL LINE(PI,YP,1024,1,0,0)                                       ASDF7738
      GO TO 40                                                          ASDF7739
   30 IF(ITYPE.EQ.2)GO TO 50                                            ASDF7740
      CALL AXIS(2.179,2.2012,17HMAGNITUDE OF H(Z),+17,5.,90.,YM(1025), ASDF7741
     XYM(1026))                                                         ASDF7742
      CALL PLOT(2.179,2.2012,-3)                                        ASDF7743
      CALL LINE(PI,YM,1024,1,0,0)                                       ASDF7744
      GO TO 40                                                          ASDF7745
   50 CALL AXIS(2.179,2.2012,22HMAGNITUDE OF H(Z) (DB),                 ASDF7746
     X+22,5.,90.,YMDB(1025),YMDB(1026))                                 ASDF7747
      CALL PLOT(2.179,2.2012,-3)                                        ASDF7748
      CALL LINE(PI,YMDB,1024,1,0,0)                                     ASDF7749
   40 CALL PLOT(-2.179,-2.2012,-3)                                      ASDF7750
      NUMPLT=NUMPLT+1                                                   ASDF7751
      RETURN                                                            ASDF7752
      END                                                               ASDF7753
                                                                        ASDF7754
      SUBROUTINE BOX(I,FLTRGN,NUMPLT)                                   ASDF7755
C*                                                                      ASDF7756
C* THIS SUBROUTINE DRAWS THE BOUNDARIES OF                              ASDF7757
C* THE PLOTTING AREAS                                                   ASDF7758
C*                                                                      ASDF7759
      DIMENSION CORNER(6,2)                                             ASDF7760
      DATA CORNER /1.,0.,13.,0.,13.,0.,1.,10.,-10.,10.,-10.,10./        ASDF7761
      CALL FACTOR(.89)                                                  ASDF7762
      CALL PLOT(CORNER(NUMPLT,1),CORNER(NUMPLT,2),-3)                   ASDF7763
      CALL PLOT(12.375,0.,+2)                                           ASDF7764
      CALL PLOT(12.375,9.5625,+2)                                       ASDF7765
      CALL PLOT(0.,9.5625,+2)                                           ASDF7766
      CALL PLOT(0.,0.,+2)                                               ASDF7767
      IF(I.EQ.0)RETURN                                                  ASDF7768
      CALL SYMBOL(7.4922,1.83,.139,14HFILTER GAIN = ,0.,+14)            ASDF7769
```

```
      CALL NUMBER(9.4297,1.83,.139,FLTRGN,0.,+5)                        ASDF7770
      RETURN                                                            ASDF7771
      END                                                               ASDF7772
                                                                        ASDF7773
```

REFERENCES

1 Blackman, R. B., _Data Smoothing and Processing_, p. 76, Addison-Wesley, 1965.

2 Chen, Chi-Tsong, _One-Dimensional Digital Signal Processing_, p. 289-331, Marcel Dekker, 1979.

3 Hamming, R. W., _Digital Filters_, p. 213-219, Prentice Hall, 1977.

4 Naval Postgraduate School, _Users Manual_, p. 3.1-4.69, 1979.

5 Oppenheim A. V. and Schafer R. W., _Digital Signal Processing_, 148-181, Prentice Hall, 1975.

6 Rabiner, L. R. and Gold, B., _Theory and Application of Digital Signal Processing_, 40-354, Prentice Hall, 1975.

7 Raney, S. D., _Using the Versatec Plotter at NPS_, 1979.

8 Stanley, W. D., _Digital Signal Processing_, pp. 88-101, Reston, 1975.

9 Tektronix, _4012 Computer Display Terminal Users Instruction Manual_, 1973.

10 Tektronix, _Plot 10 Terminal Control System User's Manual_, 1977.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department Of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor D. C. Kirk, Code 62Ki Department Of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
5.	Professor S. Parker, Code 62Px Department Of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Professor R. D. Strum, Code 62St Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	Professor Robert H. Nunn, Code 69Nm Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Ms. Jane Foust, Code 0141 Naval Postgraduate School Monterey, California 93940	1
9.	Captain Gerald S. Doyle, USA Orchard Drive Chester, New Jersey 07930	2